ALICE-TDR-19

CERN-LHCC-2015-006
November 30, 2015

Technical Design Report for the Upgrade of the Online–Offline Computing System

The ALICE Collaboration*

Copyright CERN, for the benefit of the ALICE Collaboration.
This article is distributed under the terms of Creative Commons Attribution License (CC-BY-3.0), which permits any use provided the original author(s) and source are credited.

*See Appendix D for the list of collaboration members

Scope of the document

The ALICE Collaboration is preparing a major upgrade which has been presented to the LHCC in an Upgrade Letter of Intent (LoI) and four Upgrade Technical Design Reports (TDRs) for the Inner Tracking System (ITS), the Time Projection Chamber (TPC), the Muon Forward Tracker (MFT) and the Read-out and Trigger System.

This document is the TDR for the Upgrade of the ALICE Online and Offline Computing to a new common system called O^2. The ALICE computing model, the functions of the new software framework and the new computing facility proposed to be built at the ALICE experimental area are set out below, together with the schedule of the project and the resources needed.

Executive Summary

In 2020, the ALICE experiment at CERN will start collecting data with an upgraded detector. This upgrade is based on the LHC running conditions after LS2[1] which will deliver Pb–Pb collisions at up to $\mathscr{L} = 6 \cdot 10^{27}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, corresponding to an interaction rate of 50 kHz. The ALICE goal is to integrate a luminosity of 13 nb^{-1} for Pb–Pb collisions recorded in minimum bias mode together with dedicated p–Pb and pp reference runs.

The main physics topics addressed by the ALICE upgrade are precise measurements of heavy flavour hadrons, low-momentum quarkonia and low mass di-leptons. These physics probes are characterised by a very small signal-to-background ratio requiring very large statistics. This large background makes triggering techniques very inefficient, if not impossible. In order to keep up with the 50 kHz interaction rate, the TPC will also require the implementation of a continuous read-out process to deal with event pile-up and avoid trigger-generated dead time. Compared to Runs 1 and 2, this is significantly more challenging for the online and offline computing systems. The resulting data throughput from the detector has been estimated to be greater than 1 TB/s for Pb–Pb events, roughly two orders of magnitude more than in Run 1. To minimise the cost and requirements of the computing system for data processing and storage, the ALICE Computing Model for Runs 3 and 4 is designed for a maximal reduction of the data volume read out from the detector as early as possible during the data-flow. The zero-suppressed data of all collisions will be shipped to the O^2 facility at the anticipated interaction rate of 50 kHz for Pb–Pb collisions or 200 kHz for pp and p–Pb collisions. Detector read-out will be activated either by a minimum bias trigger or in a continuous mode. The data volume reduction will be achieved by reconstructing the data in several steps synchronously with data taking. For example, the raw data of the TPC (the largest contributor to the data volume) will first be rapidly reconstructed using online cluster finding and a first fast tracking using an early calibration based on average running conditions. Data produced during this stage will be stored temporarily.

Taking advantage of the duty factor of the accelerator and the experiment, the second reconstruction stage will be performed asynchronously, using the final calibration in order to reach the required data quality. The O^2 facility will be sufficient to perform both the synchronous and asynchronous reconstruction during pp data taking. The asynchronous reconstruction will process data archived at the O^2 facility or parked at the Tier 0. However, during Pb–Pb data taking, part of the asynchronous data processing load will be shed to external sites such as Grid Tier 0 and Tier 1s that can provide archiving capability.

The O^2 facility will be a high-throughput system which will include heterogeneous computing platforms similar to many high performance computing centres. The computing nodes will be equipped with hardware acceleration. The O^2 software framework will provide the necessary abstraction so that common code can deliver the selected functionality on different platforms. The framework will also support a

[1] The terms LS1, LS2 and LS3 refer to the LHC Long Shutdowns in 2013-14 and anticipated in 2018-19 and 2023-25, during which the detector upgrades occur. Run 1, Run 2, Run 3 and Run 4 refer to the periods of data taking operation of ALICE in between these shutdowns.

concurrent computing model for a wide spectrum of computing facilities, ranging from laptops to the complete O^2 system. Off-the-shelf open source software conforming to open standards will be used as much as possible, either for the development tools or as a basis for the framework itself.

This TDR describes the ALICE computing model for Runs 3 and 4. An essential aspect is the presentation of a combined ALICE online and offline computing system which has evolved from the systems used in Runs 1 and 2. It shows in more detail how the O^2 facility can be viably implemented using proven technologies and the software framework currently under development, given a conservative but realistic budget.

A continuous evolution of the Grid has been assumed in line with Worldwide LHC Computing Grid (WLCG) projections based on a funding model for Runs 3 and 4, similar to that for Runs 1 and 2. The development schedule is compatible with the starting date for Run 3 and the projected running scenarios. Adequate safety margins are included in the schedule. Human resources with the required competences and experience are available in the institutes contributing to the project. All the institutes who have successfully designed, built and operated the ALICE DAQ, HLT and offline systems are members of the O^2 project. New institutes have joined the project, reinforcing the initial ALICE computing team.

Contents

1 Introduction **1**

1.1 Physics objectives 1

1.2 ALICE upgrade 1

1.3 Computing upgrade overview 2

1.3.1 Upgrade concept 2

1.3.2 The ALICE computing model 4

1.3.3 The O^2 software framework 4

1.3.4 The O^2 facility 4

1.3.5 The O^2 project resources 4

1.4 Document summary 5

2 Physics programme and data taking scenario **7**

2.1 Physics programme with the upgraded ALICE detector 7

2.2 Integrated-luminosity requirements for Pb–Pb and trigger strategy 8

2.3 Integrated luminosity requirements for pp reference runs 9

2.4 ALICE running scenario after Long Shutdown 2 10

3 Requirements and constraints **13**

3.1 The ALICE detector upgrade 13

3.2 Data rates, sizes and throughput 14

3.3 Detector read-out links 14

3.4 Common read-out unit 15

3.5 Local/global processing 17

3.6 Physics simulation 18

3.7 Distributed processing and analysis 18

4 Computing model **21**

4.1 Computing model parameters . . . 22
4.2 Computing and storage requirements . . . 23
4.3 Role of the O^2 facility . . . 24
4.4 Differences between pp and Pb–Pb data taking . . . 25
4.5 Roles of Tiers . . . 25
4.6 Analysis facilities . . . 27
4.7 Distributed computing . . . 28
4.8 Data management . . . 29
4.9 Replication policy . . . 29
4.10 Deletion policy . . . 29
4.11 Conditions data and software distribution . . . 30
4.12 Workflows . . . 30
4.12.1 Calibration and reconstruction . . . 30
4.12.2 Simulation . . . 31
4.12.3 Analysis . . . 31

5 O^2 architecture 33

5.1 Overview . . . 33
5.2 Data flow . . . 35
5.2.1 Data model . . . 35
5.2.2 Time Frame size . . . 36
5.2.3 Read-out links . . . 38
5.2.4 FLP and EPN . . . 38
5.2.5 The FLP-EPN network . . . 38
5.3 Data storage . . . 40
5.3.1 Condition and calibration database . . . 40
5.4 Calibration, reconstruction and data reduction . . . 41
5.4.1 Calibration and reconstruction flow . . . 41
5.4.2 Data volume reduction . . . 42
5.5 Data quality control and assessment . . . 42
5.5.1 General QC workflow . . . 42
5.5.2 Detector requirements . . . 42
5.5.3 Input and output . . . 43
5.5.4 Automation . . . 43

5.5.5 Access to QC results . . . 44
5.6 Facility control, configuration and monitoring . . . 44
5.6.1 Overview . . . 44
5.6.2 System functions . . . 45
5.7 Detector Control System . . . 46

6 Technology survey 49
6.1 Introduction . . . 49
6.2 Computing platforms . . . 49
6.2.1 Input-Output (I/O) . . . 49
6.2.2 Memory bandwidth . . . 51
6.2.3 FPGA data flow online processing . . . 52
6.2.4 CPUs . . . 52
6.2.5 Accelerator architectures . . . 53
6.2.6 Selection criteria . . . 54
6.3 High-level benchmarks . . . 54
6.3.1 FPGA based Online Data Preprocessing . . . 54
6.3.2 HLT TPC Track Finder . . . 55
6.3.3 HLT TPC Track Fitter (HLT Track Merger) . . . 56
6.3.4 ITS Cluster Finder . . . 56
6.4 Low-level benchmarks . . . 57
6.4.1 PCIe benchmark . . . 57
6.4.2 Compute benchmarks (DGEMM / Matrix lib based) . . . 58
6.4.3 memcpy() benchmark . . . 58
6.4.4 NUMA benchmark . . . 58
6.5 Performance conversion factors . . . 58
6.6 Programming models . . . 59
6.6.1 Compute-Kernels . . . 59
6.6.2 Programming languages and frameworks . . . 61
6.7 Networking . . . 62
6.8 Data storage hardware . . . 63
6.9 Technology bets . . . 64
6.10 Data storage software . . . 66
6.10.1 Cluster file system . . . 66

6.10.2 Object stores . . . 69
6.10.3 CERN EOS system - Object storage software . . . 69
6.10.4 Reliability and Recovery . . . 71
6.11 Control, configuration and monitoring . . . 71
6.11.1 DIM and SMI . . . 72
6.11.2 ZeroMQ and Boost Meta State Machine . . . 72
6.11.3 MonALISA . . . 72
6.11.4 Zabbix . . . 74

7 O^2 software design 75
7.1 Introduction . . . 75
7.2 ALFA . . . 76
7.2.1 Data transport layer . . . 76
7.2.2 Payload protocol . . . 77
7.3 Facility control, configuration and monitoring . . . 78
7.3.1 Tasks . . . 78
7.3.2 Process state machine . . . 79
7.3.3 Roles and activities . . . 79
7.3.4 Agents . . . 81
7.3.5 System monitoring . . . 82
7.4 Readout . . . 83
7.5 Data model . . . 84
7.5.1 Single data headers . . . 85
7.5.2 Multiple data headers . . . 86
7.5.3 Time Frame descriptor . . . 86
7.5.4 FLP data aggregation . . . 87
7.5.5 EPN data format . . . 88
7.6 Data quality control and assessment . . . 89
7.6.1 Data collection and QC data production . . . 89
7.6.2 Merging . . . 89
7.6.3 Automatic checks . . . 90
7.6.4 Correlation and trending . . . 90
7.6.5 Storage . . . 91
7.6.6 QC Results . . . 91

7.6.7 Flexibility . . . 91
7.6.8 Event display . . . 91
7.7 DCS-O^2 communication interfaces . . . 92

8 Physics software design 95
8.1 Calibration & reconstruction . . . 95
8.1.1 Calibration and reconstruction steps . . . 95
8.1.2 Calibration procedures . . . 97
8.1.3 Reconstruction procedures . . . 103
8.1.4 Global tracking . . . 106
8.1.5 Event extraction . . . 106
8.1.6 Computing requirements . . . 107
8.2 Physics simulation . . . 107
8.2.1 Runs 3 and 4 simulation requirements . . . 108
8.2.2 Transport code and the virtual Monte Carlo interface . . . 109
8.2.3 Detector response and digitisation . . . 109
8.2.4 Fast simulation . . . 109

9 Data reduction 111

10 O^2 facility design 115
10.1 Introduction . . . 115
10.2 O^2 Facility . . . 115
10.3 Power and cooling facilities . . . 118
10.4 Demonstrators . . . 119
10.4.1 Data transport . . . 119
10.4.2 The Dynamic Deployment System . . . 120
10.4.3 Future demonstrators . . . 121
10.5 Computing system simulation . . . 121
10.5.1 Simulation of the network needs . . . 121
10.5.2 Simulation of the buffering needs . . . 122
10.5.3 Data storage simulation . . . 123
10.5.4 Simulation of the system scalability . . . 124

11 Project organisation, cost estimate and schedule 127

11.1 Project management 127
11.2 O^2 Project 127
11.3 Schedule 129
11.4 Responsibilities and human resources estimates 129
11.5 Cost estimates and spending profile 131
11.6 Production and procurement 131
11.7 Risk analysis 131
11.8 Data preservation 132
11.9 Open access 132
11.10 O^2 TDR editorial committee 133
11.11 O^2 TDR authors 133

Appendices

A Glossary **137**

B Integrated luminosity requirements for proton–proton reference runs **142**

C Tools and procedures **145**
C.1 Evaluation procedure 145
C.2 Selected tools 146
C.2.1 Issue/Bug tracking system: JIRA 146
C.2.2 Version control system: Git 147
C.2.3 Website creation tool: Drupal 148
C.2.4 Source code documentation tool: Doxygen 149
C.2.5 Software build system: Cmake 149
C.2.6 Computing simulation tool: Omnet++ and the Monarc simulation tool 150
C.3 Policies 150
C.3.1 C++ Coding conventions 150

D The ALICE Collaboration **153**

References **159**

List of Figures **165**

List of Tables **169**

Chapter 1

Introduction

ALICE (A Large Ion Collider Experiment) [1] is a general purpose, heavy ion collision detector at the CERN LHC [2]. It is designed to study the physics of strongly interacting matter, and in particular the properties of Quark-Gluon Plasma (QGP), using proton-proton, nucleus-nucleus and proton-nucleus collisions at high energies. The ALICE experiment will be upgraded during the Long Shutdown 2 (LS2) in order to exploit the full scientific potential of the future LHC.

1.1 Physics objectives

The physics programme of the ALICE experiment after its upgrade is discussed in the Letter of Intent (LoI) [3, 4]. The study of the QGP in the second generation of LHC heavy ion research following LS2 will focus on rare probes and the study of their coupling with the medium and hadronisation processes. These include heavy flavour particles, quarkonium states, real and virtual photons, jets and their correlations with other probes. A description of the main physics items of the programme with the upgraded detector is reported in Chap. 2, along with the required measurements and the projected performance estimated from simulation studies.

The LHC running conditions after the LS2 foresee Pb–Pb collisions at the centre-of-mass energy per nucleon–nucleon collision $\sqrt{s_{\mathrm{NN}}} = 5.5$ TeV with an instantaneous luminosity up to $\mathscr{L} = 6 \cdot 10^{27}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ corresponding to the hadronic interaction rate of 50 kHz. The upgraded ALICE detector will be able to read out all interactions and achieve the goal of collecting 13 nb^{-1} of Pb–Pb collisions (10 nb^{-1} at the nominal value of the ALICE solenoid magnetic field and 3 nb^{-1} at a reduced value of the field) [3]. Compared to the original programme of the experiment, this integrated luminosity represents an increase of a factor of ten in the data sample for rare triggers and of a factor of one hundred for the minimum bias data sample. The physics programme also requires a reference pp sample corresponding to an integrated luminosity of 6 nb^{-1} at the same centre-of-mass energy as for Pb–Pb, $\sqrt{s} = 5.5$ TeV, and a sample of p–Pb collisions corresponding to 50 nb^{-1} (the p–Pb centre-of-mass energy will be either 5.5 TeV, as for Pb–Pb, or 8.8 TeV, the maximum LHC energy for this colliding system). Data from proton–proton collisions will also be collected at the nominal LHC energy, $\sqrt{s} = 14$ TeV, during the first year after the upgrade and then for about two months every year, prior to the heavy ion run.

1.2 ALICE upgrade

The ALICE detector upgrade includes:

- A new, high-resolution, low material Inner Tracking System (ITS) [5];

- An upgrade of the Time Projection Chamber (TPC) [6] consisting of the replacement of the wire chambers with Gas Electron Multiplier (GEM) detectors and new continuous read-out electronics;
- The addition of a Muon Forward Tracker (MFT) [7];
- An upgrade of the read-out electronics of several detectors: Muon CHamber System (MCH), Muon IDentifier (MID), Transition Radiation Detector (TRD) and the Time-Of-Flight detector (TOF) [8];
- A new Central Trigger Processor (CTP) and a new Fast Interaction Trigger (FIT) detector [8].

The existing detectors that will continue to operate after LS2 are listed in Chap. 3.

The topic of this Technical Design Report is the replacement of the Data AcQuisition (DAQ) [9], the High-Level Trigger (HLT) and the offline systems by a new common Online-Offline computing system (O^2).

1.3 Computing upgrade overview

The ALICE upgrade addresses the challenge of reading out and inspecting the Pb–Pb collisions at rates of 50 kHz, sampling the pp and p–Pb at up to 200 kHz. The tracking precision of the experiment will be improved at both central and forward rapidity. This will result in the collection and inspection of a data volume of heavy ion events roughly 100 times greater than that of Run 1.

1.3.1 Upgrade concept

The ALICE computing upgrade concept consists of transferring all detector data to the computing system. The data volume reduction will be performed by processing the data on the fly in parallel with the data collection and not by rejecting complete events as do the high-level triggers or event filter farms of most high-energy physics experiments. The O^2 system will perform a partial calibration and reconstruction online and replace the original raw data with compressed data. The online detector calibration and data reconstruction will therefore be instrumental in keeping the total data throughput within an envelope compatible with the available computing resources.

The functional flow of the O^2 system includes a succession of steps as shown in Fig. 1.1 The data will be transferred from the detectors either in a continuous fashion or by using a minimum bias trigger. The continuous read-out of some detectors is a substantial change from current practice. The data are not delimited by a physics trigger but is composed of several constant data streams that will be transferred to the computing system. Dedicated time markers will be used to chop these data flows into manageable pieces called Time Frames (TF) and the LHC clock will be used as a reference to synchronise, aggregate and buffer the data.

A first local calibration and detector specific pattern recognition will be carried out with a high degree of parallelism due to the local and independent nature of the data. Some of the detector raw data will already be replaced at this stage by the results of a first local processing step. For example: the TPC raw data will be replaced by the results of the cluster finding. A second step of data aggregation is performed to assemble the data from all the detector inputs. A global calibration and data processing is performed synchronously with the data taking, typically associating the clusters to tracks. The results are stored in the O^2 farm if the capacity allows it or are parked in the Tier 0.

A final processing step is then performed asynchronously before permanently storing the reconstructed events. This final step may use computing resources from the Grid to absorb the peak needs beyond the capacity of the O^2 system. The reconstructed events will then be available for analysis on the Grid.

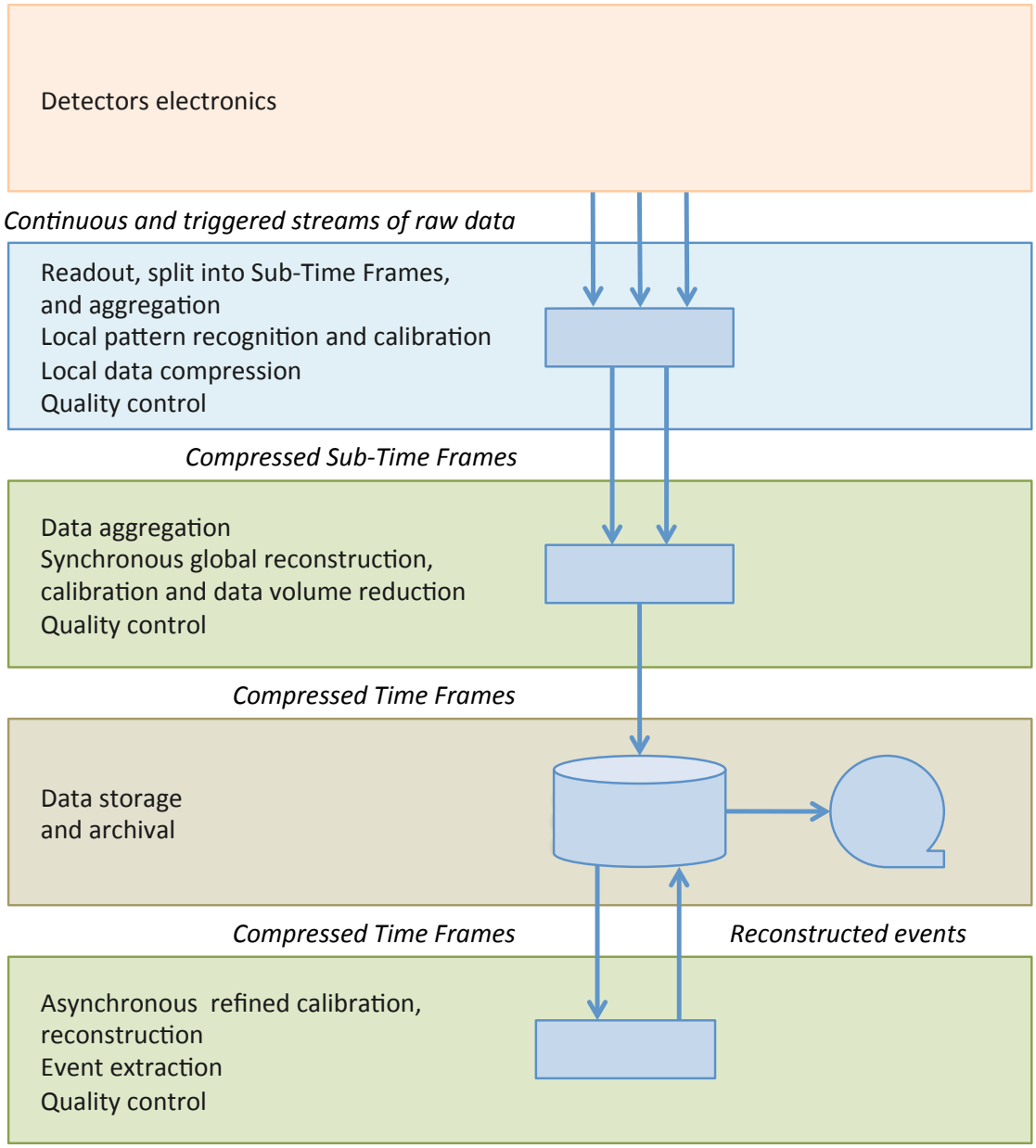

Figure 1.1: Functional flow of the O^2 computing system.

The data processing steps performed during data taking will keep the option open for subsequent calibrations of the most critical parameters to protect the physics results in case of a software error or operational mistake.

There will be substantial detector pile-up due to the anticipated collision rate of 50 kHz in Pb–Pb and 200 kHz in pp. The event identification will only be possible at the very end of the reconstruction by associating the tracks and secondary vertices to a particular bunch crossing. At this point, the fully reconstructed data will be stored at the experimental area ready for archiving.

The main role of the O^2 system will be to perform detector calibration and data reconstruction concurrently with data taking. The integration of online and offline data processing will require a common O^2 software framework and a common computing facility dedicated to both data collection and processing.

1.3.2 The ALICE computing model

Data processing will be shared between the O^2 facility and the Grid. During data taking, the O^2 facility will carry out the synchronous fast reconstruction and, within limits of its capacity, the asynchronous reconstruction storing the output in the local O^2 data store. The O^2 facility will be used for subsequent iterations on calibration, raw data processing and extraction of analysis object data.

The ALICE computing model for Run 3 assumes a continuous evolution of the Grid according to WLCG projections based on a funding model similar to the one used during Runs 1 and 2. In order to be compatible with the expected Grid evolution and to minimise the cost of the offline data processing, there will be a drastic reduction in the volume of data collected from the experiment as early as possible in the process.

1.3.3 The O^2 software framework

Building on the experience accumulated during the design and operation of the online and offline systems during Run 1 and Run 2, the O^2 software framework will implement a distributed, parallel and staged data processing model. It is designed from the start to combine all the computing functionalities needed in a HEP experiment: detector read-out, event building, data recording, detector calibration, data reconstruction, physics simulation and analysis. It can be tailored to different configurations by the adaptation or addition of detector specific algorithms and to specific computing systems. The O^2 framework will support the evolution from today's scheme of single threaded applications into several small components communicating by messages and running concurrently and in a transparent manner on several cores of the same or different nodes. The O^2 framework will also support the use of hardware accelerators where available.

1.3.4 The O^2 facility

The O^2 facility, located at the experimental area at Point 2, will also provide the interfaces with the Grid and notably the permanent data store at Tier 0 as well as for the data processing in all centres. The O^2 facility will therefore be part of the overall ALICE computing model; it will have a data storage system with a storage capacity large enough to accommodate a large fraction of the output data of a full year's data taking.

Outside of data taking periods, the O^2 facility will allow reconstruction and provide the necessary capability for detector tests and commissioning.

1.3.5 The O^2 project resources

The O^2 facility, as the other ALICE upgrade projects scheduled for Run 3 will be largely built on the existing infrastructure at Point 2.

Institutes that have participated in the successful design, building and operation of the online and offline systems since the start of the ALICE experiment are participating in the O^2 project. The adhesion of additional institutes has reinforced the project to give it the right size, capacity and competences necessary to realise the O^2 system. So far, the project comprises 31 institutes for a total of over 90 participants.

The development schedule is compatible with LS2 from mid 2018 till the end of 2019 and Run 3 starting in 2020.

The project has established a realistic construction budget based on the estimated needs and a conservative evolution of computing technology until deployment, which is scheduled from 2017 until 2020.

1.4 Document summary

The physics programme addressed by the ALICE upgrade, the running scenarios with the integrated luminosity requirements in pp and Pb–Pb beams, the trigger strategy and the corresponding data taking scenarios are discussed in Chap. 2.

The requirements and constraints imposed by the physics programme are presented in Chap. 3 including: the architecture requirements for detector read-out; the triggering and the resulting event rates and data throughput; the data processing and physics simulation needs for pp and Pb–Pb collisions.

Chapter 4 presents an update of the computing model following the ALICE upgrade. The sharing of the data processing load between the O^2 facility and the Grid is presented together with the usage by different categories of data processing: calibration, synchronous and asynchronous reconstruction, simulation and analysis.

Chapter 5 expands on the architecture of the O^2 data processing system. The data model and the architecture of the main hardware and software components of the O^2 facility and software framework are described.

The current state of computing technology is considered in Chap. 6. In particular, the existing computing platforms and programming models are reviewed together with their benchmarking using actual ALICE algorithms.

The design of the dataflow and control software, the physics software, and the data reduction are presented in Chap. 7, Chap. 8 and Chap. 9 respectively. Chapter 10 is entirely dedicated to hardware considerations for the design of the O^2 facility.

The project organisation, the cost estimate and the project schedule are considered in Chap. 11. This chapter also includes a list of the participating institutes and a preliminary view of how responsibilities can be shared out.

Chapter 2

Physics programme and data taking scenario

The physics requirements that define the ALICE data taking strategy and schedule for Runs 3 and 4 are presented in this chapter. The physics programme of the ALICE experiment after its upgrade is discussed in detail in the Letter of Intent [1] and its Addendum on the Muon Forward Tracker [2]. See Sec. 2.1 below for a brief summary.

The targeted integrated luminosity is 13 nb^{-1} in Pb–Pb collisions at the LHC design energy $\sqrt{s_{\mathrm{NN}}} =$ 5.5 TeV, of which 3 nb^{-1} are to be collected with a lower magnetic field in the ALICE solenoid. The integrated luminosity value for Runs 3 and 4 represents an increase of a factor of ten with respect to the rare-trigger sample expected for Run 2 and of a factor of 100 with respect to the minimum bias sample. As summarised in Sec. 2.2, this integrated luminosity is required for many of the central measurements of the physics programme, from low-p_{T} heavy flavour and charmonium production to low-mass dilepton production. Due to the very low signal-to-background ratio that is expected in the low-p_{T} region for most of the relevant channels (e.g. $\mathrm{D}^0 \to \mathrm{K}^-\pi^+$ and $\Lambda_{\mathrm{c}}^+ \to \mathrm{pK}^-\pi^+$), the rate of collisions of interest, containing a signal candidate, will be of the same order as the interaction rate. As a result of this, it has been concluded that online event filtering is not a viable strategy for these low-p_{T} measurements [3]. The method consists in recording all Pb–Pb interactions. Section 2.2 also reiterates this conclusion, on the basis of updated performance estimations from the Technical Design Report of the ITS Upgrade [4].

Requirements for pp reference data are discussed in Sec. 2.3. The requirement, first introduced in the LoI [1], for a sample at the same centre-of-mass energy as the Pb–Pb sample, $\sqrt{s} = 5.5$ TeV, with an integrated luminosity of about 6 pb^{-1} is confirmed by estimations for a number of reference measurements. The requirements for data at the full LHC energy of 14 TeV have not yet been worked out with good accuracy. Therefore, a minimal scenario of data taking at this energy is assumed for the moment. The integrated luminosity requirement for p–Pb collisions is 50 nb^{-1} [1]. A possible implementation of the running schedule is presented in Sec. 2.4.

2.1 Physics programme with the upgraded ALICE detector

The study of the strongly-interacting state of matter in the second generation of the LHC heavy ion studies following LS2 will focus on rare probes, and the study of their coupling with the medium and hadronisation processes, including heavy flavour particles, quarkonium states, real and virtual photons, jets and their correlations with other probes. The cross sections of all these processes are significantly larger at the LHC than at previous accelerators. In addition, the interaction of heavy flavour probes with the medium is better controlled theoretically than the propagation of light partons. All these investi-

gations involve soft momentum scales, and thus benefit from the ALICE detector strengths: excellent tracking performance in a high-multiplicity environment and particle identification over a large momentum range. In most of these studies, the azimuthal anisotropy of different probes will be measured. Major highlights of the proposed programme will focus on the following physics topics:

- Thermalisation of partons in the QGP, with focus on the charm and beauty quarks. Heavy-quark azimuthal-flow anisotropy is especially sensitive to the partonic equation of state. Ultimately, heavy quarks might fully equilibrate and become part of the strongly-coupled medium.
- Low-momentum quarkonium dissociation and, possibly, regeneration pattern, as a probe of deconfinement and an evaluation of the medium temperature.
- The production of thermal photons and low-mass dileptons emitted by the QGP. This should allow the assessment of the initial temperature and the equation of state of the medium, as well as investigation of the chiral nature of the phase transition.
- The study of the in-medium parton energy-loss mechanism. This provides, on the one hand, a testing ground for the multi-particle aspects of QCD and, on the other hand, a probe of the density of the QGP and of its response to energy loss. The relevant observables are: jet structure, jet–jet and photon–jet correlations; jet correlations with high-momentum identified hadrons; heavy flavour particle production in jets. In particular, it is crucial to characterise the dependencies of energy loss on the parton colour-charge, mass, and energy, as well as on the density of the medium.
- The search for heavy nuclear states such as light multi-Λ hyper-nuclei ${}^{5}_{\Lambda\Lambda}$H, bound states of ($\Lambda\Lambda$) or the H dibaryon; a systematic study of the production of light nuclei, anti-nuclei and of (Λn) bound states.

2.2 Integrated-luminosity requirements for Pb–Pb and trigger strategy

The physics performance for an integrated luminosity of 10 nb^{-1} at full magnetic field in the ALICE solenoid (+3 nb^{-1} with reduced field of $B = 0.2$ T) is presented in the ALICE Upgrade LoI [1] and its Addendum [2] for all measurements listed in the previous section, with updates for heavy flavour measurements at central rapidity, low-mass dielectrons and hypernuclei in the ITS Upgrade TDR [4].

The requirement for an integrated luminosity of $(10+3)\,\mathrm{nb}^{-1}$ is motivated mainly by the following performance figures.

Heavy flavour measurements:

- The nuclear modification factor R_{AA} and elliptic flow v_2 of strange-charmed mesons ($\mathrm{D_s}$) down to a transverse momentum p_{T} of at least 2 GeV$/c$ with a statistical precision better than 10% for both observables. This will allow a precise comparison of strange and non-strange charm meson dynamics;
- Λ_{c} baryon R_{AA} and v_2 down to 2 GeV$/c$ and 3 GeV$/c$, respectively, with a precision of about 20% and baryon/meson ratio for charm ($\Lambda_{\mathrm{c}}/\mathrm{D}$) down to 2 GeV$/c$ with the same precision. This will allow to address the charm quark hadronisation mechanisms at low and intermediate momentum;
- R_{AA} and v_2 of beauty-decay particles via non-prompt D^0, non-prompt J/ψ and beauty-decay leptons (the two latter at both central and forward rapidity) down to 1–2 GeV$/c$ with precisions from a few percent to 10%. This will allow a detailed assessment of the b quark transport properties in the medium;

- B meson fully-reconstructed decays ($B^+ \to \overline{D^0}\pi^+$) down to 3 GeV/$c$ with a precision of about 10%. This will provide an important direct measurement of beauty production;
- Λ_b baryon production for $p_T > 7$ GeV/c. This will be a unique measurement in heavy ion collisions and should allow for the determination of the nuclear modification factor of the beauty baryon, which is sensitive to the b quark hadronization mechanism.

Charmonium measurements:

- R_{AA} of J/ψ down to $p_T = 0$ with statistical precision better than 1%, at both central and forward rapidity;
- R_{AA} of $\psi(2S)$ down to $p_T = 0$ with precision of about 10%, at both central and forward rapidity;
- v_2 of J/ψ down to $p_T = 0$ with precision of about 0.05 (absolute uncertainty), at both central and forward rapidity;

these measurements will allow a detailed investigation of the mechanisms of dissociation and regeneration for charmonium states in the deconfined medium.

Low-mass dileptons: the additional sample of 3 nb^{-1} with a reduced magnetic field value (0.2 T) in the central barrel is essential for low-mass dielectron analysis to obtain the projected precision of about 10% on the slope of the high-invariant-mass region and of about 10% on the dielectron elliptic flow. This measurement will make it possible to assess the time-evolution of the thermal radiation emitted by the hot medium.

All these analyses (with the exception of the exclusive reconstruction of beauty hadron decays) are characterised by a very small signal-to-background ratio, implying that there is a signal candidate in essentially all Pb–Pb collisions. This means that it is not possible to use dedicated triggers to select collisions online for offline analysis. Instead, a minimum bias trigger must be used and all collisions recorded. In the following, this point is reiterated using an update of the estimates presented in the Conceptual Design Report of the ITS Upgrade [3].

Table 2.1 reports the expected signal per event (S/ev), signal-to-background ratio (S/B) and number of background candidates per event[1] (B'/ev) for central Pb–Pb collisions at $\sqrt{s_{NN}} = 5.5$ TeV. The signal per event takes into account the branching ratios and the reconstruction and selection efficiencies (see Tables 2.2 and 2.3 of Ref. [3]). Since S/B is very small, the total number of candidates (signal+background) is essentially the same as the number of background candidates.

For the D^0 and D_s^+ mesons there are more than 7 candidates per event. For the case of the Λ_c, also considering a minimum p_T threshold at 2 GeV/c (the lowest accessible p_T according to studies presented in [4]), due to the small S/B ratio, there are about 7 candidates per event. Therefore, it can be concluded that an online event selection is not adequate for low-p_T heavy flavour measurements. All Pb–Pb interactions have to be recorded and inspected offline.

2.3 Integrated luminosity requirements for pp reference runs

The integrated luminosity required for the pp reference runs was estimated on the premise that the statistical uncertainty on the pp reference is lower than that of the Pb–Pb measurement. The pp uncertainty

[1] The number of background candidates per event, for a given particle, is estimated as the background in the broad invariant mass range that is necessary to fit the invariant mass distribution (e.g. $\pm 12\,\sigma$, where σ is the invariant mass resolution for that particle). Therefore, $B'/\text{ev} = (S/\text{ev})/(S/B') = 4 \cdot (S/\text{ev})/(S/B)$, where S/B is the signal-to-background in $\pm 3\,\sigma$ of the invariant mass distribution.

Table 2.1: Estimated signal per event (S/ev), signal-to-background ratio (S/B) and number of background candidates per event (B'/ev, see text for details) for central Pb–Pb collisions at $\sqrt{s_{\rm NN}} = 5.5$ TeV. The reported values are derived in the Conceptual Design Report and in the TDR for the ITS upgrade, where the geometrical selections are also described [3, 4].

Analysis	S/ev [3]	S/B [4]	B'/ev
$\mathrm{D}^0 \to \mathrm{K}^-\pi^+$	$7.6 \cdot 10^{-3}$	10^{-2}	3.0
$\mathrm{D}_\mathrm{s}^+ \to \mathrm{K}^+\mathrm{K}^-\pi^+$	$2.3 \cdot 10^{-3}$	$< 2 \cdot 10^{-3}$	> 4.6
$\Lambda_\mathrm{c}^+ \to \mathrm{pK}^-\pi^+$	$6.5 \cdot 10^{-4}$	$< 10^{-4}$	> 26
$\Lambda_\mathrm{c}^+ \to \mathrm{pK}^-\pi^+$ ($p_\mathrm{T} > 2$ GeV/c)	$3.7 \cdot 10^{-4}$	$2 \cdot 10^{-4}$	7.4

was required to be $\sqrt{2}$ times smaller than the Pb–Pb uncertainty. Thus, the combined relative statistical uncertainty in, for example, a ratio of yields in Pb–Pb and in pp is at most 20% larger than the Pb–Pb uncertainty. Since the relative statistical uncertainty is the inverse of the statistical significance $1/\mathscr{S} = \sqrt{S+B}/S$, the requirement is: $\mathscr{S}_{\rm pp} = \sqrt{2} \cdot \mathscr{S}_{\rm Pb-Pb}$.

The requirements for some of the measurements described in the previous section are derived in appendix B. The results are summarized in Tab. 2.2, for the centre-of-mass energy 5.5 TeV. All values are in the range 3–6 pb^{-1}.

Table 2.2: Summary of integrated luminosity requirements for pp collisions for heavy flavour and quarkonium measurements (see Appendix B for details).

Measurement	$L_{\rm int}$ for pp at $\sqrt{s} = 5.5$ TeV, pb^{-1}
D^0	6
Λ_c^+	5
$(\mathrm{B} \to)\mathrm{J}/\psi \to \mathrm{e}^+\mathrm{e}^-$ (central rapidity)	5
$\mathrm{J}/\psi \to \mathrm{e}^+\mathrm{e}^-$ (central rapidity)	5
$\psi(2\mathrm{S}) \to \mathrm{e}^+\mathrm{e}^-$ (central rapidity)	5
$\mathrm{J}/\psi \to \mu^+\mu^-$ (forward rapidity)	5
$\psi(2\mathrm{S}) \to \mu^+\mu^-$ (forward rapidity)	5

2.4 ALICE running scenario after Long Shutdown 2

Table 2.3 shows the running scenario presented in the ALICE Upgrade LoI [1], indicating the corresponding number of recorded collisions. Estimates for the required integrated luminosity in pp collisions at $\sqrt{s} = 14$ TeV are not yet available. Therefore, a baseline scenario is considered in which data at this energy are collected during six periods of a few weeks prior to each yearly heavy ion run. In addition, the recommissioning of the full upgraded detector will be performed during the full first year of operation after LS2 although with limited efficiency.

Table 2.3: ALICE running scenario for the LHC Runs 3 and 4 (taken from [1], with the addition of pp collisions at $\sqrt{s} = 14$ TeV).

Year	System	$\sqrt{s_{\mathrm{NN}}}$ (TeV)	L_{int} pp: (pb^{-1}) p–Pb: (nb^{-1}) Pb–Pb: (nb^{-1})	$N_{\mathrm{collisions}}$
2020	pp	14	0.4	$2.7 \cdot 10^{10}$
	Pb–Pb	5.5	2.85	$2.3 \cdot 10^{10}$
2021	pp	14	0.4	$2.7 \cdot 10^{10}$
	Pb–Pb	5.5	2.85	$2.3 \cdot 10^{10}$
2022	pp	14	0.4	$2.7 \cdot 10^{10}$
	pp	5.5	6	$4 \cdot 10^{11}$
2025	pp	14	0.4	$2.7 \cdot 10^{10}$
	Pb–Pb	5.5	2.85	$2.3 \cdot 10^{10}$
2026	pp	14	0.4	$2.7 \cdot 10^{10}$
	Pb–Pb	5.5	1.4	$1.1 \cdot 10^{10}$
	p–Pb	8.8	50	10^{11}
2027	pp	14	0.4	$2.7 \cdot 10^{10}$
	Pb–Pb	5.5	2.85	$2.3 \cdot 10^{10}$

Chapter 3

Requirements and constraints

This chapter summarises the computing requirements of the ALICE experiment for the Runs 3 and 4 physics programme. These requirements cover all the phases from data taking, processing, analysis to physics simulation. They will be addressed by the O^2 facility and the Grid resources.

Section 3.1 presents a short summary of the detector system after the upgrade. Section 3.2 envisages the typical data taking parameters, necessary for estimating the dimensions of data flow, data processing and data storage. The system is scalable to address larger requirements but the project size and budget are based on the baseline scenario presented below. A short review of detector read-out is included in Sec. 3.4. It includes the upgraded and new detectors, the ones which will be equipped with new electronics and those which will keep their existing electronics. Sections 3.5 and 3.6 are dedicated to an estimate of the computing requirements for the data processing needed for data volume reduction before it is archived and for physics simulation. The requirements for distributed processing and analysis on the Grid are discussed in Sec. 3.7.

3.1 The ALICE detector upgrade

To meet the challenges of the physics programme outlined in Chap. 2 the ALICE detector needs to be upgraded to enhance its low-momentum vertexing and tracking capability, and to allow data taking at substantially higher rates. The new detector system is described in detail in the ALICE Upgrade documents: the Letter of Intent [1] and its Addendum on the Muon Forward Tracker [2]; the four Upgrade Technical Design Reports (Inner Tracking System [3], Time Projection Chamber [4], The Muon Forward Tracker [5] and The Read-out and Trigger System [6]). The upgraded ALICE detector will include:

- A new, high-resolution, low-material ITS, based on 7 layers of monolithic silicon pixel detectors replacing the current Silicon Pixel Detectors (SPD), Silicon Drift Detectors (SDD), and Silicon Strip Detectors (SSD). The ITS will be able to provide read-out rates of 100 kHz for Pb–Pb and 200 kHz for pp collisions.
- An upgraded TPC. The current TPC is based on a gated read-out with wire chambers. Due to the gating grid closure time, it is limited to read-out rates of less than 3.5 kHz. In order to allow continuous read-out operation of 50 kHz Pb–Pb collisions, the wire chambers will be replaced with GEM detectors. Digitised and timestamped data will be pushed to the online system. A triggered mode will be available for commissioning and calibration.
- The addition of a MFT, located in the forward region between the ITS and the front absorber. It consists of several discs of monolithic silicon pixel detectors in the acceptance of the Muon Spectrometer (MCH and MID, see below).

- The MCH consisting of a sequence of 5 wire chamber stations in the forward region of ALICE. The detector implementation will not be modified, but the entire read-out electronics will be replaced to achieve a read-out rate of 100 kHz, two orders of magnitude larger than at present. The dead time free read-out will support a self-triggered continuous as well as a triggered mode.
- The MID, the evolution of the present Muon Trigger system will consist of two stations of two planes of single-gap Resistive Plate Chamber (RPC) detectors are located 16 m and 17 m from the interaction point. All events will be read and the data will be used online for hadron rejection.
- The TRD, using tracklets to reduce the data volume. As the front-end electronics will not support multi-event buffers only 78 % of the events are read out at 50 kHz Pb–Pb collisions .
- A new FIT detector providing the minimum bias trigger for the experiment. It will replace the current forward detectors (V0, T0, FMD). It consists of Cherenkov and scintillator detectors.
- An upgrade of the read-out electronics of the TOF detector increasing the maximum read-out rate to 200 kHz for Pb–Pb events.
- The Zero Degree Calorimeter (ZDC) with upgraded read-out electronics to accept the higher interaction rate of 100 kHz .
- The Electro-Magnetic Calorimeter (EMC) and the Photon Spectrometer (PHS), which have been upgraded for 50 kHz operation during LS1.
- The High Momentum Particle Identifier (HMP), which will not be modified and will therefore be capable of reading only 2.5 kHz events.
- The ALICE Cosmic Ray Detector (ACO), which is already capable of a read-out rate of 100 kHz and will not be modified.
- The CTP, that provides trigger and timing distribution to the detectors. It will be upgraded to support the required interaction rate.

3.2 Data rates, sizes and throughput

Table 3.1 shows the maximum detector read-out rate, as well as the average data throughput and data size per interaction (or per trigger, for those being triggered at lower rate) at the detector for Pb–Pb collisions at 50 kHz.

As can be seen from the table, the main contributors are: TPC (92.5%), ITS (3.6%), TRD (1.8%) and MFT (0.9%). The other detectors contribute for the remaining 1% of the total data volume. The total data size per interaction is approximately 23 MB.

As described in Chap. 5, data are grouped for processing in time windows which duration spans over several thousands of interactions. The peak data size per interaction values are therefore not detailed here, as they average out.

3.3 Detector read-out links

The upgraded ALICE trigger system supports both continuously read-out and triggered detectors. Not all sub-systems will be capable of reading the full event rate. These detectors will therefore be read out whenever they are not busy; their data will be merged with the data from the other detectors in the O^2 system.

[1] The TRD accepted rates are 38.5, 62.5, and 90.9 kHz for respectively 50, 100 and 200 kHz interaction rates.

Table 3.1: Detector parameters: maximum read-out rate, data rate and data size. The data rate and data size are estimated for Pb–Pb interactions at a rate of 50 kHz. The numbers have been extracted from published sources of information where references available and from draft documents otherwise.

Detector name	Maximum read-out rate (kHz)	Data rate for Pb–Pb collisions at 50 kHz (GB/s)	Average data size per interaction or trigger at the detector (MB)
ACO [7]	100	0.014	0.00028
CPV	50	0.9	0.018
CTP [6]	200	0.02	0.0004
EMC [6]	42	4.0	0.08
FIT [6]	100	0.115	0.023
HMP [6, 7]	2.5	0.06	0.024
ITS [3]	100	40	0.8
MCH [6]	100	2.2	0.04
MFT [5]	100	10.0	0.2
MID [6]	100	0.3	0.006
PHS [6]	42	2.0	0.04
TOF [6]	200	2.5	0.05
TPC [4]	50	1012	20.7
TRD [6]	90.9[1]	20	0.5
ZDC [6]	100	0.06	0.0012
TOTAL		1095	22.5

Two different types of read-out links will be used to transport data from the detectors to the O^2 facility. Some detectors will continue to use their Run 1 or Run 2 Detector-Specific Read-Out (DSRO) electronics and still be read out with the Detector Data Links 1 and 2 (DDL1 and DDL2) [8, 9]. The detectors with upgraded electronics will use the GigaBit Transceiver (GBT) [10].

DDL1 is clocked at 2.125 Gb/s and the DDL1 Source Interface Unit (SIU) is implemented as a radiation-tolerant daughter card plugged on to the DSRO systems. The DDL2 SIU is implemented as an Intellectual Property (IP) core and can be clocked at 4.25 or 5.3125 Gb/s according to the capabilities of the detector electronics to which it is connected. PCI-express based Common Read-Out Receiver Cards (C-RORCs), used during Run 2, will interface up to 6 DDL1s or DDL2s to the O^2 facility. The data receiving PCs of the O^2 computing farm, the First Level Processors (FLPs), described in more detail in Sec. 5.2.4, will host up to two C-RORCs.

3.4 Common read-out unit

For the detectors using the GBT, the Common Read-Out Unit (CRU) acts as the interface between the DSRO electronics, the O^2 facility, the Detector Control System (DCS) (via the O^2 facility), as well as the CTP and the Trigger, Timing, and clock distribution System (TTS). One CRU will interface up to 24 GBT links and one FLP can host a maximum of two CRUs. Figure 3.1 shows the general ALICE read-out scheme with the three variations of read-out. The CRUs are based on high performance Field Programmable Gate Array (FPGA) processors equipped with multi gigabit optical inputs and outputs. Depending on detector specifications, detector data sent to the CRU are multiplexed, processed and formatted. The CRU on-detector interface is based on the GBT and an optical versatile link protocol and its components. Furthermore, the CRU is capable of multiplexing the data, trigger and control signal over the same GBT.

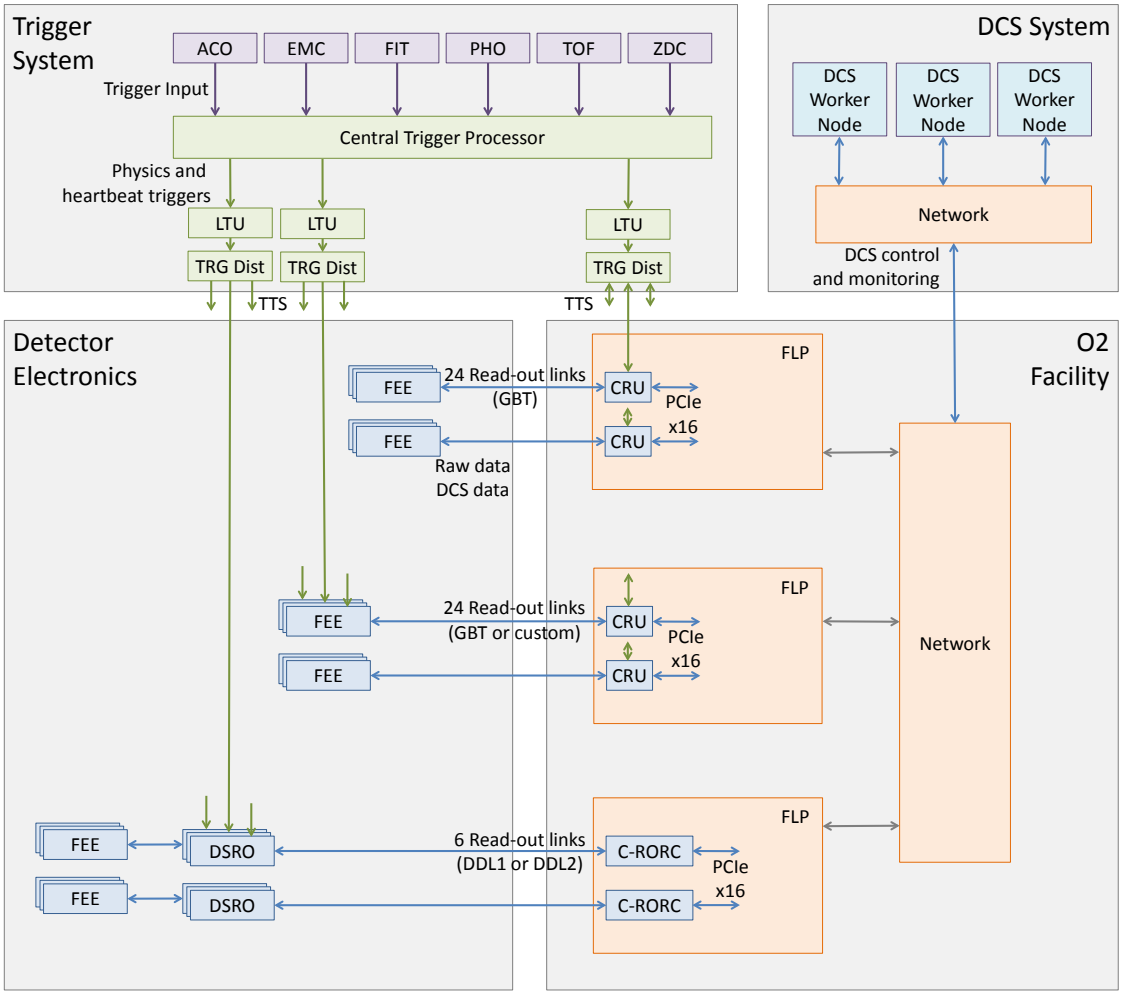

Figure 3.1: Detector read-out and interfaces of the O^2 system with the trigger, detector electronics and DCS.

The CRU FPGA will also be used for detector functions such as the on-the-fly cluster finder for the TPC data. The cluster finder will extract the clusters from the raw data, compute their properties and replace the raw data with the list of clusters with their properties. Figure 3.2 shows the block diagram of the CRU with the parts common to all detectors (TTS interface, CRU control, GBT interface and the PCIe interface to the FLP) and the detector specific logic.

The FLPs, and therefore the CRUs, are located in a counting room close to the surface while the trigger system and the detectors are in the experimental cavern. This results in two different approaches for the trigger distribution depending on the maximum delay for delivering the trigger signal to the detector: either the latency is not critical and the trigger is transmitted to the CRU which forwards it over the GBT, or it is critical and the trigger signal is transmitted directly from the Local Trigger Unit (LTU) to the DSRO electronics.

The number and type of link and read-out board needed by the detectors are summarized in Tab. 3.2 based on a CRU interfacing 24 links to the FLP.

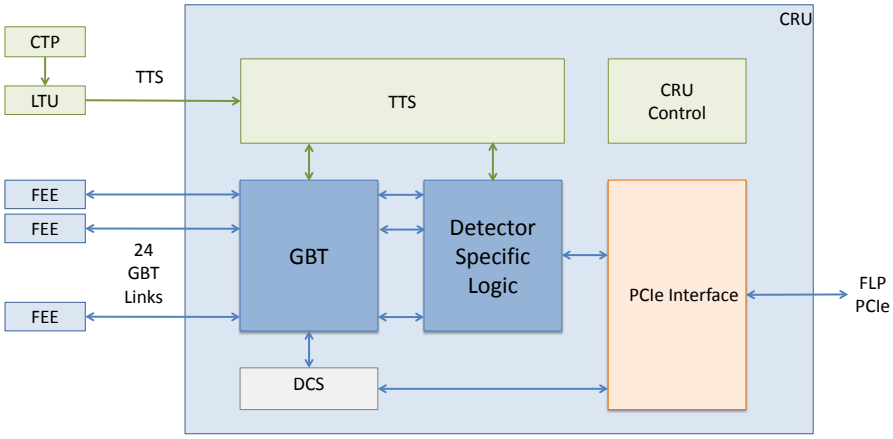

Figure 3.2: Block diagram of the Common Read-out Unit.

Table 3.2: Detector links and read-out boards used to transfer the data from the detectors to the O^2 system [6].

	Detector	Link type	Number of links DDL1	DDL2	GBT	Read-out board type	Number of boards C-RORC	CRU
	ACO	DDL1	1			C-RORC	1	
	CPV	DDL1	6			C-RORC	1	
	CTP	GBT			14	CRU		1
	EMC	DDL2		20		C-RORC	4	
	FIT	DDL2		2		C-RORC	1	
	HMP	DDL1	14			C-RORC	4	
short	ITS	GBT			495	CRU		23
	MCH	GBT			550	CRU		25
	MFT	GBT			304	CRU		14
	MID	GBT			32	CRU		2
	PHS	DDL2		16		C-RORC	4	
	TOF	GBT			72	CRU		3
	TPC	GBT			5832	CRU		324
	TRD	Custom			1044	CRU		54
	ZDC	GBT			1	CRU		1
	Total		21	38	8344		15	447

3.5 Local/global processing

The overall processing schema includes the following tasks:

- Calibration and reconstruction of the raw data coming from the ALICE detector.
- Monte Carlo (MC) simulation and reconstruction of simulated data. The amount of fully simulated data is expected to be factor two greater than in Run 1. The results of the full (slow) simulation can be used either directly or to tune fast simulation tools (see Sec. 3.6).
- Analysis. The analysis of reconstructed data typically take place on the Grid or at dedicated analysis facilities (see Sec. 3.7).

The *local processing* takes place directly where the data links deliver continuous raw data flow (e.g. from parts of ITS or TPC) or triggered data (from parts of the other detectors). The term local processing underlines the fact that only partial information from the ALICE detector is available at this stage. One of the main goals of the local processing is data reduction, in particular for the TPC (see Chap. 9).

During *global processing*, the information of the full ALICE detector is available as input. Global processing includes detector reconstruction, calibration and quality control. Different global processing strategies are possible: standalone tracking in the barrel detectors and in the muon arm; track matching between the detectors, using fast seeding algorithms and track propagation; detector specific calibration algorithms. The details of the reconstruction and calibration steps are described in Sec. 8.1.1. The main goals of the global processing are data compression, physics quality calibration and reconstruction.

3.6 Physics simulation

The requirements for the number of MC events can be estimated from the need to determine the reconstruction efficiency of rare signals (charm/beauty-hadrons, high-p_{T} jets, ...) up to the highest reachable p_{T}. Moreover, it can be assumed that trivial variance reduction techniques, like equalizing number of events over p_{T} bins, will be employed. With respect to Run 1, increasing the number of events in data by a factor of 1000 will increase the p_{T} reach approximately by a factor of 5, justifying an equal increase in the number of simulated events. In general, the availability of larger data sets triggers new and more differential analysis that may need higher dimensional correction maps. To account for this possibility, the number of simulated events needs to be further increased by an order of magnitude. For Run 1 about 10^7 central Pb–Pb collisions have been simulated. This leads to an estimate of $5 \cdot 10^8$ events for Run 3, 0.5% of the number of data events. Assuming the retention of the current practice of increasing the number of rare signals per event by at least an order of magnitude, the generated number of signal events becomes commensurable with data. The corresponding number of pp MC events amounts to $5 \cdot 10^{10}$.

Table 3.3: Number of simulated events for different systems.

System	Events
pp	$5 \cdot 10^{10}$
Pb–Pb	$5 \cdot 10^{8}$
p–Pb	$5 \cdot 10^{9}$

3.7 Distributed processing and analysis

Grid resources

The WLCG infrastructure is used by the LHC experiments for all large-scale storage and computational needs, including storage and replication of raw data and its processing, Monte-Carlo simulation, individual and organised user analysis. This infrastructure represents a significant investment in hardware, in the forms of large, medium and small scale regional and institute computing centres and networks, as well as in support personnel. Over the years, the WLCG software has undergone consistent development and tuning to meet the evolving requirements of the user community. The resources deployed in WLCG have also increased, following the needs of the experiments. In general, the resources growth has been proportional to the size of the individual collaborations, with ALICE using approximately 20% of the globally available CPU, disk and tape. The ALICE-specific increase is shown in Table 3.4 (CPU, disk and tape) during Run 2 of LHC operation. The table includes the projected values at the start of Run 3. The projections are based on a “flat” hardware investment scenario, adopted already during Run 1, where the growth is based on improvement in CPU and storage technologies. The values given for the period 2015-2017 are taken from the ALICE requirements documents approved by the Computing Resources

Scrutiny Group. These are evaluated and updated every calendar year and the table values reflect both the averages taken from the documents as well as the historical resources evolution.

Table 3.4: Grid resources evolution 2015-2019. The year-on-year increase is shown in brackets in the corresponding cells.

Year	CPU (kHEPSpec)	Disk (PB)	Tape (PB)
2015	495	49	26
2016	615 (+25%)	59 (+20%)	37 (+42%)
2017	725 (+18%)	68 (+16%)	45 (+22%)
2018	870 (+20%)	93 (+20%)	54 (+20%)
2019	1045 (+20%)	112 (+20%)	54 (no increase)

The growth of the Grid resources and capabilities is expected to cover the needs for compressed data storage, as outlined in Chap. 4. In addition, the Grid will continue to be the primary platform for MC simulations and for end-user analysis. Various physics simulation scenarios are given in Sec. 3.6, with an emphasis on fast and parametric models, which will require substantially less CPU power than the full MC simulation used during Runs 1 and 2. The specific repartition of work done on the Grid is given in Sec. 4.5.

Chapter 4

Computing model

The ALICE Computing Model for Runs 3 and 4 is driven by the need to reduce the data volume to the maximum possible extent in order to minimise the storage cost and requirements of the computing resources needed for data processing while minimising the impact on physics performance.

In spite of significant network improvements over the past decades the ALICE experience during Run 1 shows that certain workflows such as I/O limited data analysis cannot be run efficiently in a fully distributed computing environment such as Grid (see Fig. 4.1). By reducing the data volume early in the processing chain, the data that needs to be moved between the components of the system is minimised, thereby reducing the impact on the network.

This is the rationale for changing the current computing model based on the Grid as an abstraction of a computing infrastructure which is, in principle, uniform and capable of executing any kind of job, anywhere, for a new approach where certain sites and facilities are dedicated to specific activities. In this model, the bulk of data is produced and stored locally for subsequent re-processing and, in its digested and reduced form, pre-placed at those specific locations that are dedicated to and optimised for analysis activity.

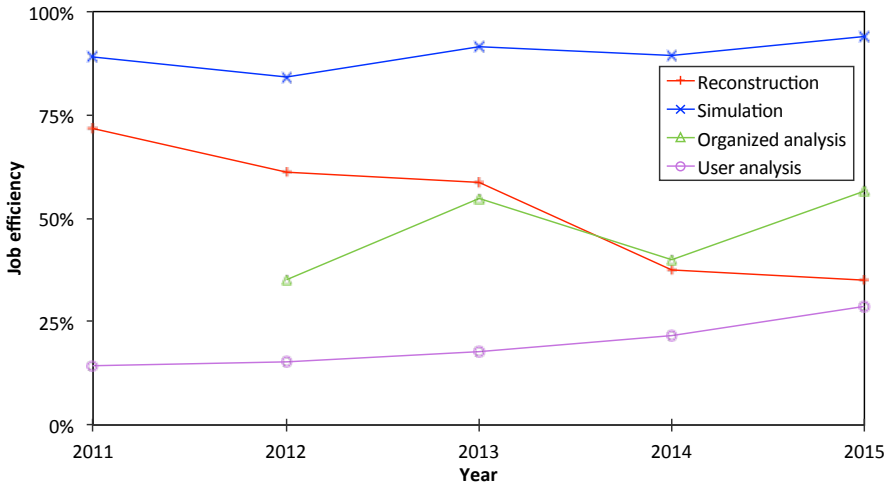

Figure 4.1: Job efficiency (CPU/Wall Clock) for various ALICE workflows running on the Grid during Run 1.

Table 4.1 summarizes the components of the computing model in terms of computing facilities, their primary roles and capabilities.

Table 4.1: Components of the system.

Facility	Function
O^2	ALICE Online-Offline Facility at LHC Point 2. Its primary task during data taking is to run the online reconstruction in order to achieve maximal data compression. It provides storage sufficient for 2/3 of the compressed data and runs the calibration and reconstruction tasks.
Tier 0	CERN Computer Centre facility providing CPU, storage and archiving resources. Here reconstruction and calibration tasks are carried out on a portion of the archived compressed data, plus simulation if required.
Tier 1	Grid site connected to Tier 0 with high bandwidth network links (at least 100 Gbps) providing CPU, storage and archiving resources. It runs the reconstruction and calibration tasks on its portion of archived compressed data with simulation if needed.
Tier 2	Regular grid site with good network connectivity running simulation jobs.
AF	Dedicated Analysis Facility of High Performance Computing (HPC) type that collects and stores physics analysis data produced elsewhere and runs the organised analysis activity.

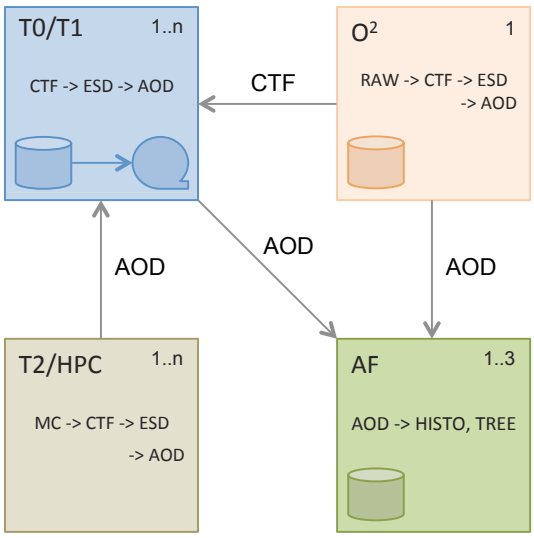

Figure 4.2: Data flow between components of the system.

4.1 Computing model parameters

This chapter summarises the parameters of the ALICE computing model in terms of data types, event rates and data flow. The data types that constitute the Runs 3 and 4 computing model are shown in Tab. 4.2.

Table 4.3 shows the size of the different data types used in the O^2 system.

Table 4.2: Data Types in Run 3.

Acronym	Description	Persistency
RAW	Raw data as it comes from the detector	Transient
STF	Sub-Time Frame containing raw data from a single FLP for a period of time $\approx 20\,\text{ms}$.	Transient
CTF	Compressed Time Frame containing processed raw data of all active detectors for a period of time $\approx 20\,\text{ms}$. In the case of TPC, clusters not belonging to tracks or identified as background are rejected and the remaining information is compressed to the maximum. Once written, CTF becomes read only data.	Persistent
ESD	Event Summary Data. Auxiliary data to CTF containing the output of the reconstruction process linking the tracks and clusters.	Temporary
MC	Monte Carlo. Simulated energy deposits in sensitive detectors. Removed once the reconstruction of MC data is completed.	Transient
AOD	Analysis Object Data containing the final track parameters in a given vertex and for a given physics event. AODs are collected on dedicated facilities for subsequent analysis.	Persistent
MCAOD	Analysis Object Data for a given simulated physics event. Same as AOD with addition of kinematic information that allows comparison to MC. MCAODs are collected on dedicated facilities for subsequent analysis.	Persistent
HISTO	The subset of AOD information specific for a given analysis. Can be generated during analysis but needs to be offloaded from the Grid.	Temporary

Table 4.3: Relative size of different data types.

Data type	Event size (kB)	Number of versions	Tape copy
CTF (pp)	50	1	Yes
CTF (p–Pb)	100	1	Yes
CTF (Pb–Pb)	1600	1	Yes
ESD	15% of CTF	1	No
AOD	10% of CTF	2	Yes
MC	100% of CTF	1	No
MCAOD	30% of CTF	2	Yes
HISTO	1% of ESD	1	No

4.2 Computing and storage requirements

This section estimates the resources required to process and store the data of Runs 3 and 4. The number of reconstructed collisions and the average data size per interaction are taken from Tab. 2.3 and Tab. 3.1 respectively. Table 4.4 summarizes the estimated storage requirements, supposing an overall compression factor of 14 after the data reconstruction in Pb–Pb (see Chap. 9) and a factor of 10 in pp and p–Pb.

Table 4.4: Number of reconstructed collisions and storage requirements for different systems and scenarios.

Year	System	Collisions	Storage CTF (PB)	Storage Calibration (TB)	Storage ESD/AOD (PB)	Required CPU seconds (single CPU core)
2020	pp	$2.7 \cdot 10^{10}$	1.5	5	0.6	$1.7 \cdot 10^{10}$
	Pb–Pb	$2.3 \cdot 10^{10}$	37	23	15	$2.8 \cdot 10^{11}$
2021	pp	$2.7 \cdot 10^{10}$	1.5	5	0.6	$1.7 \cdot 10^{10}$
	Pb–Pb	$2.3 \cdot 10^{10}$	37	23	15	$2.8 \cdot 10^{11}$
2022	pp	$4.3 \cdot 10^{11}$	23	76	9.2	$2.7 \cdot 10^{11}$
2025	pp	$2.7 \cdot 10^{10}$	1.5	5	0.6	$1.7 \cdot 10^{10}$
	Pb–Pb	$2.3 \cdot 10^{10}$	37	23	15	$2.8 \cdot 10^{11}$
	pp	$2.7 \cdot 10^{10}$	1.5	5	0.6	$1.7 \cdot 10^{10}$
2026	Pb–Pb	$1.1 \cdot 10^{10}$	18	11	7.2	$1.3 \cdot 10^{11}$
	p–Pb	$1.0 \cdot 10^{11}$	10	20	4.0	$7.2 \cdot 10^{10}$
2027	pp	$2.7 \cdot 10^{10}$	1.5	5	0.6	$1.7 \cdot 10^{10}$
	Pb–Pb	$2.3 \cdot 10^{10}$	37	23	15	$2.8 \cdot 10^{11}$

Table 4.5: Number of simulated events and storage requirements for different systems.

System	Events	Storage MCAOD (PB)
pp	$5 \cdot 10^{10}$	0.75
central Pb–Pb	$5 \cdot 10^{8}$	1.2
p–Pb	$5 \cdot 10^{9}$	0.15

4.3 Role of the O^2 facility

The ALICE Computing Model for Run 3 assumes that in the first stage of data processing all (RAW) data coming from the detector will be synchronously reconstructed in the O^2 facility, including cluster finding and fast tracking based on approximate calibration. The resulting output in Compressed Time Frames (CTF), containing the history of possibly thousands of overlapping events, will then be stored locally. Up to this point, data compression will reduce the data volume by a large factor (see Chap. 9) relative to that of the raw data coming from the detectors. From here on, the Time Frames are immutable and eligible for archiving.

To ensure sufficiently fast processing, it is assumed that at least part of the O^2 facility will have to be equipped with hardware accelerators such as Graphics Processing Units (GPUs) and FPGAs and that the software will need to be developed to take full benefit of the hardware capabilities.

The second stage in this process is an asynchronous reconstruction procedure that will take place in the available part of the O^2 facility during data taking or across the whole facility whenever data taking stops. After this stage the aim is to have a final calibration and reconstruction of the required quality. Figure 4.3 shows the processing steps and data flow within the O^2 facility.

The calibration/reconstruction procedures will produce an auxiliary ESD containing information about tracks found during the reconstruction and will reference the clusters in the immutable Time Frame file. It is assumed that at most two iterations of reconstruction/calibration will be needed in order to obtain data of sufficient quality for physics analysis. The size of ESD files is estimated to be at most 15% of

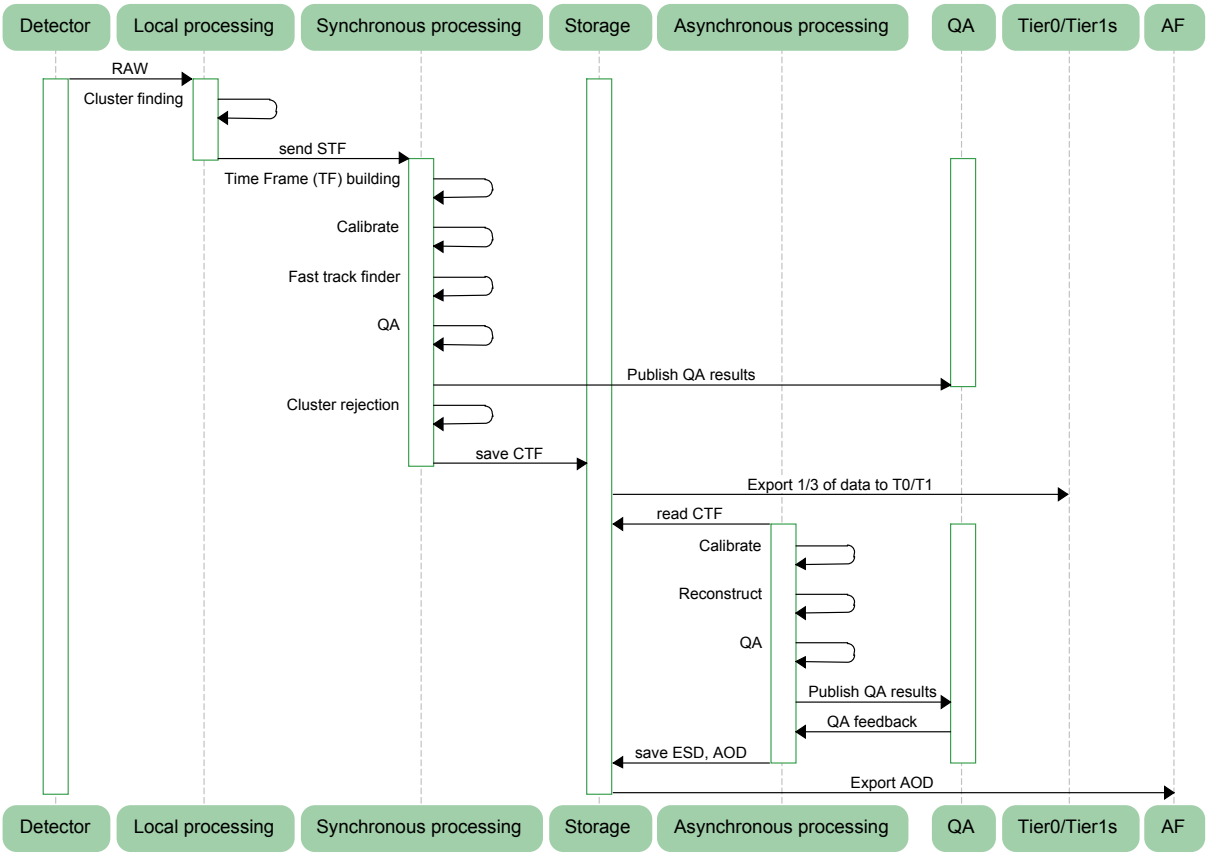

Figure 4.3: O^2 processing flow.

CTF data.

4.4 Differences between pp and Pb–Pb data taking

During pp data taking, full synchronous and asynchronous reconstruction will be possible in the O^2 facility. However, during Pb–Pb data taking, it will be necessary to delay or offload part of the asynchronous data processing to well connected Grid sites. Once the entire dataset is reconstructed with a given software version then AOD re-filtering will generate datasets suitable for analysis.

4.5 Roles of Tiers

As shown in Fig. 4.2, the roles of the (Worlwide LHC Computing Grid) WLCG Tiers will remain similar but not identical to those of the current ALICE computing model. While the intention is to retain the ability to run any task anywhere, the main difference compared to Run 1 is the specialist role allotted to different sites, as described below.

During data taking, the O^2 facility will carry out synchronous fast reconstruction and, within the limits of its capacity, asynchronous reconstruction storing the output in the local data store. The O^2 farm should be able to keep up with processing and will store up to 2/3 of all CTF data during Pb–Pb data taking and 50% during pp periods. The O^2 facility will be used for subsequent iterations of calibration, reconstruction and AOD re-filtering.

It is expected that Tier 0 will continue to evolve towards a private cloud offering infrastructure as a

service. It will complement the O^2 computing resources and take part in the asynchronous reconstruction. Tier 0 will backup on tape all data from the O^2 disk buffer as shown in Fig. 4.4.

All CTF data stored in the O^2 facility (2/3 of the total CTF volume) will be archived on Tier 0 tape. In addition to its role in prompt reconstruction, Tier 0 will run the simulation. Simulation type jobs will run as a backfill at times when there is no calibration or reconstruction activity. The resulting AODs will be archived and sent for subsequent processing to dedicated Analysis Facilities (AFs).

In general, the AODs will be systematically sent to the AFs to run organised analysis. During the latter process, a further skimming of datasets will be possible to generate micro AODs containing n-tuples or histograms suitable for processing off the Grid.

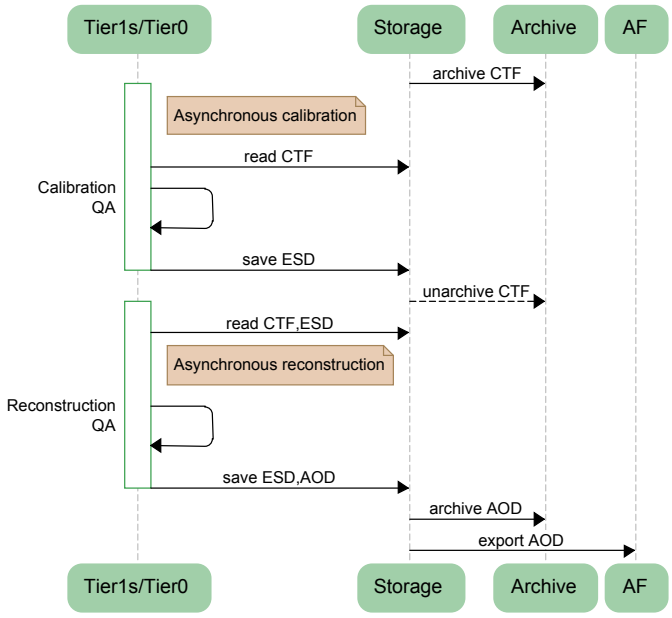

Figure 4.4: Tier 0/Tier 1 processing flow.

Tier 1 sites will contribute to asynchronous reconstruction of the remaining 1/3 of CTF data (see Fig. 4.4). These sites will need to have, on average, an attainable 20 Gbps network connectivity, roughly corresponding to the expected ALICE share of the network bandwidth on Tier 1s. Once data are moved to the remote site, they should be reconstructed in exactly the same way as in the O^2 facility and finally archived to tape. Similarly to Tier 0, Tier 1 sites will contribute to subsequent calibration and reprocessing as well as to other activities such as simulation when there is no other activity.

The asynchronous processing (calibration and re-reconstruction) of data collected during one data taking period has to complete before the next data taking period starts. At that point, all CTF data will be removed from O^2 and Tier 1s disk buffers to make room for new data. Unprocessed data, if any, will remain parked on tape until the next long shutdown that will open a window of opportunity for reprocessing.

From the data management perspective, in the current ALICE computing model, each Tier 2 storage element appears individually. In the future, they will be assigned to regional groups including at least one Tier 1 site to form a cloud of sites responsible for storing, archiving and subsequent data processing (see Fig. 4.5). These sites should be close enough geographically and in network terms (latency, bandwidth) to ensure cross-site data access with minimal performance penalties. Within such regional clouds, tasks of organised simulation will be carried out with the associated reconstruction of MC data followed by

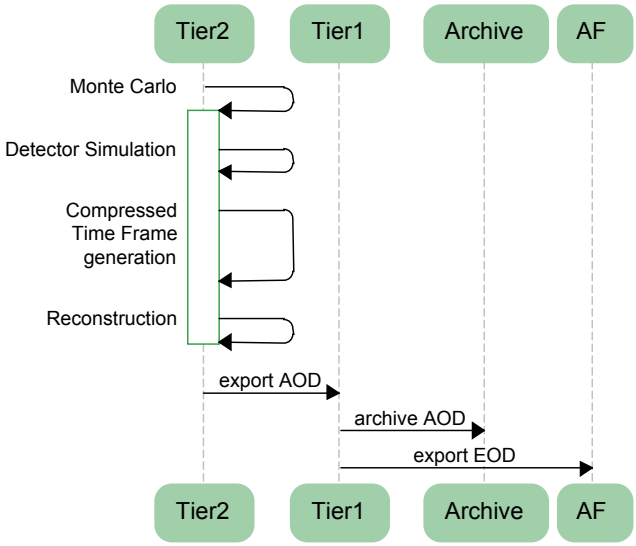

Figure 4.5: Tier 2 simulation processing flow.

the re-filtering and production of AODs. As in all other cases, the resulting AODs will be transferred to AFs for subsequent analysis.

4.6 Analysis facilities

In the present ALICE computing model there are no explicit Tier 3 sites but several national or regional AFs providing a service processing subsets of ALICE data of specific interest to certain communities.

This model will be optimised: a small number (2-3) of similar AFs will be identified to provide a general service to all ALICE users and will also run organised analysis following the well established analysis train model (see Sec. 4.12.3). The sites concerned must have very good network connectivity and a large, efficient local disk buffer 5 PB dedicated to the storage of an entire set of active ALICE AODs with the associated CPU capacity for efficient processing of AODs several times per day. Possible candidates for a new AF are

- A fraction of an existing Tier 1 centre, upgraded, with high-performance data storage and the required CPU capacity dedicated to its new role.
- An existing Tier2 centre providing an equivalent amount of storage and CPU capacity.

In addition, it is expected that more resources could be provided like:

- A new purpose built facility that fulfils the requirements for an AF, dedicated to ALICE or shared with another experiment.
- A share of an HPC facility with the required storage and CPU capacity.

Today's analysis trains have from 5 to 100 different wagons and run on datasets of up to 50 TB. Each train runs twice a week and is synchronised to the deployment of the analysis code. Analysis trains average about 5,000 concurrent running jobs with peaks at 20,000. As no significant increase in the number of

physics working groups is foreseen, it is estimated that the number of analysis trains and wagons will not change much for Run 3. Analysis trains need on average 5 MB/s per job slot to be reasonably efficient. This number together with the cluster file system performance of the train will define how many job slots can be used efficiently. Given the performance of large HPC installations, it should be possible to serve 20,000 job slots from the local cluster file system at an aggregate throughput of 100 GB/s. The AF will therefore have to be able to digest more than 4 PB of AODs in a 12 hours period, corresponding to the desired turnaround time.

The main benefit of a new scheme would be in optimized I/O and faster post processing steps (merging of the output files) that should become more predictable and efficient when done within a single site than in a fully distributed way as we do it today. At present, the processing time is essentially determined by the slowest site that participates in a given analysis train. In the current Grid setup we can already identify the sites where analysis jobs run with efficiency over 70% while the average across all sites is about 50%. It is between those sites that we would like to identify possible candidate sites for the future AF role.

We argue that a purpose-build analysis facility, with optimal internal storage and network architecture, would achieve even higher efficiency for analysis jobs and thus maximize both the analysis workflow and the resources utilization.

4.7 Distributed computing

The Workload Management and Data Management systems in ALICE are presently based on AliEn, a set of middleware tools and services developed by the Collaboration and used for massive Monte Carlo event production since the end of 2001 and for user data analysis since 2005. The AliEn job management services compose a three-layer lightweight system that leverages the deployed resources of the underlying WLCG infrastructures and services, including the local variations, such as European Grid Infrastructure (EGI), Nordic Data Grid Facility (NDGF) and Open Science Grid (OSG). The three layers include: the AliEn Central Services that manage the whole system and distribute the workload; the AliEn Site Services that manage the interfacing to local resources and Grid Services. Most of the Workload Management related services revolve around the Task Queue (TQ), a central database that keeps track of all jobs submitted to the system and their current execution status. A number of Job Optimisers scan the TQ re-arranging the jobs in order to enforce fair share and priority policies and splitting them into sub-jobs according to the location of the required input data or user-defined criteria. Following the late binding approach, the JobAgents that are started on the Worker Nodes download and execute the actual payload from the central TQ.

ALICE has used AliEn for the distributed production of Monte Carlo data, reconstruction and analysis at over 80 sites. So far more than 280 million jobs have been successfully run worldwide from the TQ, resulting in the currently active volume of 14 PB of data.

While the Grid has served the purpose well for the processing of Run 1 data for LHC experiments, the sites are gradually embracing the model of private clouds and the virtualisation of their physical computing resources. This is the process that is already underway at CERN and at several big WLCG Tier 1 sites; it is not expected that this will introduce a big shift in the Grid distributing computing paradigm already in place. Using the virtualisation technology, each experiment will be able to construct its own Grid, spanning various Tiers and regional clouds, while continuing to use most of the existing distributed computing infrastructure already built and tested during Runs 1 and 2.

In particular, very complex existing workflows and production management tools with associated job and system monitoring will be kept, as they do not show scalability issues. Similarly, the job handling model based on pilot jobs and late binding to the real job has proven to be the solution for the large scale

computing problems of the LHC experiments. This will continue to be used with possible simplifications coming from the use of virtual machines that provide natural isolation between users of the physical resources. The scalability of central services that handle job distribution will be assessed and improved if necessary but the basic concept will be retained. In a limiting case, given that the sites in our new computing model will be assigned the particlular roles, the system can be easily scaled by adding another instance of the central services dedicated to the reconstruction jobs.

In summary, the existing distributed computing infastucture can be reconfigured and scaled to realize the new computing model illustrated in Fig. 4.2 requiring only a minimal development effort.

4.8 Data management

Data management is a highly important issue common to clouds and other parts of the distributed computing infrastructure. In Run 3, it will have to scale to much higher levels in terms of number of files, overall size and I/O performance. It is vital that this topic is addressed together with other experiments and CERN/IT. The ALICE approach would be to evolve the current data management model based on a network of xrootd (ref) Storage Elements to a generalised global namespace based on the EOS (ref) system, developed by CERN/IT and currently used to manage about 100 PB of Run 1 data for all four large LHC experiments. This system is itself based on the next iteration of xrootd technology and allows for gradual and non-disruptive transition. Its implementation is a goal for the coming years.

The necessary generalisations of the EOS system in terms of a global namespace that would permit the replacement of the File Catalog are in the early stages of development. It is encouraging that initial benchmarks demonstrate scalability with no performance degradation when managing the target of more than 10^9 files. The present size of the EOS disk cache at CERN is already comparable to that of the future ALICE O^2 facility proving its scalability to the requirements.

4.9 Replication policy

At present, the raw data replication model in ALICE calls for two replicas: one full set at Tier 0 and one distributed set at the Tier 1s supporting ALICE. The Tier 1s replication fraction is proportional to the amount of tape resources pledged by the centres. Due to a substantial increase in data volume, in Run 3, there will be only one instance of each raw data file (CTF) stored on disk with a backup on tape. During p-p data taking the data size of O^2 disk buffer should be sufficient to accommodate data from the entire period. As soon as it is available, the CTF data will be archived to the Tier 0 tape buffer. During Pb-Pb data taking, the data are replicated with the maximum speed allowed by fair-share of the network to the Tier 1s, as well as by the maximum rate supported by the Tier 1 tape systems. It is expected that 1/3 of data will be transferred and archived at Tier 1s by the end of data taking. Upon archiving, CTF data will remain on disk until subsequent calibration and asynchronous reconstruction steps are completed. In case of a complete data loss, the files will be recalled from the Tier 0 tape.

4.10 Deletion policy

With the exception of raw data (CTF) and derived analysis data (AOD), all other intermediate data created at various processing stages is transient (removed after a given processing step) or temporary (with limited lifetime). Given the limited size of the disk buffers in O^2 and Tier 1s, all CTF data collected in the previous year, will have to be moved before the new data taking period starts. This constraint effectively limits the asynchronous processing stage to a maximum of 9 months in the year for the final calibration and re-reconstruction. All data not finally processed during this period will remain parked on tapes until the next opportunity for re-processing arises: LS3.

The final product of the reconstruction and simulation steps are AODs. In case of reconstruction they will be backed up on tape at Tier 0 and Tier 1s. As the simulation will predominantly run on Tier 2 sites, the AODs generated on them will be sent to an associated Tier 1 site for archiving and removed from the local storage at Tier 1. In all cases, the resulting AODs will be sent to AFs for subsequent analysis. While the disk buffer at AFs is expected to be substantial, it will again be limited and strict policies will be put in place to remove unused datasets from this buffer. This will be helped by a further refinement of a popularity service which will add the possibility of a fine-grained data removal for active datasets with low level access. If it is needed, the AODs removed from this buffer can be recalled from tape at Tier 1s and the analysis can be repeated there or they can be again sent to AFs.

4.11 Conditions data and software distribution

Efficient distribution of new software releases throughout the Grid is essential. To meet this requirement the present CernVM File System (CVMFS) will continue to be used. CVMFS can also distribute ALICE conditions data, therefore lowering the impact on data management components. This option is currently being evaluated for Run 2 within the existing ALICE Offline framework and if it proves successful, it may be used for Run 3. Alternatively, a solution based on FairRoot parameter manager will be investigated. This approach offers an abstraction of the storage layer that can be realised as ROOT or ASCII files (possibly stored in CVMFS), SQL database or a key value store (such as Riak, Memcached, Redis) that give a ready to use data replication and synchronisation across nodes or data centres.

4.12 Workflows

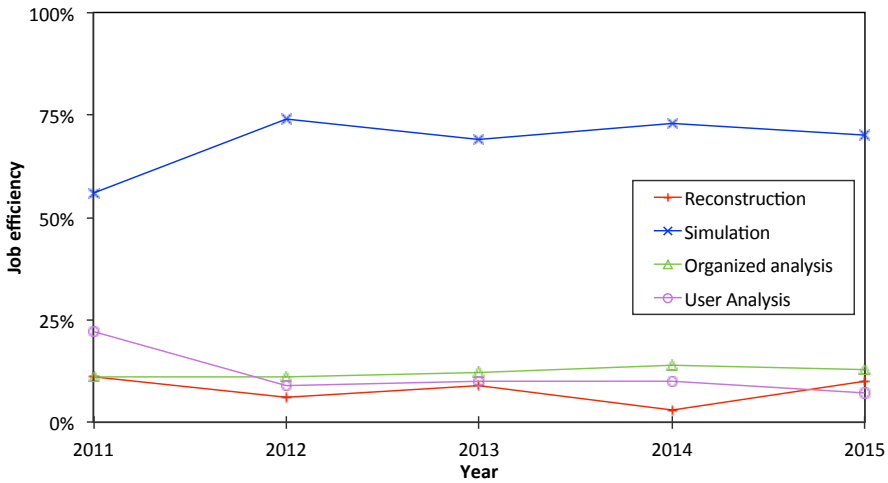

Figure 4.6: Relative share of various ALICE workflows running on the Grid during Run 1.

4.12.1 Calibration and reconstruction

During Run 1 the ALICE detector calibration was carried out in both Online (HLT) and Offline environments (Grid). This was an iterative procedure that required a lot of time and needed human intervention. The reconstruction was performed on the Tier 0 and Tier 1s, depending on the data location and processing power availability. Given that in Run 3 the first stage reconstruction needs to be done in the O^2 facility for better data compression and maximum volume reduction, the calibration and reconstruction become closely coupled and an essential part of the functionality provided by the O^2 facility. The calibration and reconstruction procedures are detailed in Sec. 8.1.

During Run 3, Tier 0 and Tier 1s will receive up to 1/3 of the data stream (CTF). For this reason, subsequent calibration and reconstruction processing will follow the same model as in the O^2 facility.

4.12.2 Simulation

So far, simulation activities have accounted for 70% of ALICE computing needs (see Fig. 4.6). While specific measures, such as use of parametrised and fast Monte Carlos, will ensure that future CPU needs for simulation will be kept to a minimum, they will continue to require a large share of resources. The estimated fraction of resources needed to simulate required number of events in Run 3 based on present computing time per event is 50%.

In addition, a great number of simulation jobs put a heavy load on the central services as each job is accounted for and monitored separately.

For Run 3 an alternative approach is being considered for bulk simulation campaigns. Dedicated simulation facilities will be implemented to handle job distribution and accounting internally while acting as data sources for the rest of the infrastructure, providing only the feed of simulated files. This model could be particularly suitable for interfacing potentially large High Performances Computing facilities that could contribute opportunely large numbers of CPU cycles. However, it is expected that the bulk of simulation jobs will run on Tier 2 sites.

As the output in form of AOD files will be sent to the AFs for subsequent analysis, it is planned to rebalance CPU/disk requirements on Tier 2s in favour of CPU. This will give an increase in the number of CPU cores available for simulation, within the limits of fixed budget.

4.12.3 Analysis

In Run 1 and in the present ALICE computing model there are two distinct types of analysis: user level analysis and scheduled analysis. They differ in their data access pattern, in the storage and registration of the results and in the frequency of changes in the analysis code.

The user level analysis is focused on a single physics task and typically is based on filtered data from scheduled tasks. Usually, physicists develop the code using a small sub-sample of data, changing the algorithms and criteria frequently. The analysis macros and software are tested many times on relatively small data volumes of both experimental and Monte-Carlo data. The output is often only a set of histograms. Such tuning of the analysis code can be done on a local data set or on distributed data using Grid tools. The final version of the analysis is eventually submitted to the Grid and will access large portions or even the totality of the AODs.

While both types of analysis were allowed and supported in Run 1, the user level analysis has proved to be less efficient by far. In Run 3 it is planned to replace this option completely by organised or scheduled analysis.

Scheduled analysis typically uses all the available data from a given period. The AOD files, read during scheduled tasks, can be used for several subsequent analyses, or by a class of related physics tasks that are combined in what are known as ”analysis trains”.

Analysis trains group the analyses of different users together that process the same dataset with the following advantages:

- The total overhead resource usage that is associated with the execution of each analysis code separately is significantly reduced. This includes: job management; input and output file management; job-start overhead and the resources consumed for the decoding of the input data;
- Users have to execute much fewer Grid operations themselves and so save time;

- Automatic systems deal with failures and merging, reducing the total turnaround time to less than that achievable by an individual user;
- Tests before the submission of an analysis train spot wrong configurations or faulty code, reducing the number of job failures.

The composition, testing and submission of trains are steered with a dedicated web front-end that allows users to configure the specific analysis code that they would like to run on certain data sets. In addition, operators can use the front-end to test train compositions to evaluate resource consumption per train wagon (CPU time and memory) and to issue warnings in case of memory leaks or excessive output. The analysis train model worked well during Run 1 and it is planned to build on its success improving its efficiency and making it the only way for doing analysis over the entire data samples in Run 3.

To help individual physicists carry out exploratory work, the scheduled analysis tools will be improved to facilitate skimming and extracting of the subsets of data for processing on local resources.

Chapter 5

O^2 architecture

5.1 Overview

The data flow and processing pipeline of the O^2 system is shown in Fig. 5.1. The O^2 architecture has been designed to support both online synchronous data reduction and asynchronous and iterative data processing.

Data are produced by the detectors in continuous or triggered read-out mode, synchronised by the trigger system. The FLPs read out the raw data samples over the GBT based Frontend links and DDLs optical links. Several streams may be aggregated on each FLP and buffered in memory. These nodes achieve a first data reduction and compression factor of 2.5 (see Chap. 9) by performing the local data processing tasks on data fragments, without needing to have the full detector data set (e.g. TPC cluster finding).

The continuous streams of data samples are split into data frames, using as a reference clock arbitrary Heartbeat triggers embedded in the raw data streams. The frames are accumulated during a time period of the order of 20 ms, to minimise incomplete data at their boundaries for the collisions producing tracks spanning across the frame boundaries, as detailed in Sec. 5.2.2. All FLPs produce a Sub-Time Frame (STF), which could be empty for those FLPs receiving data from triggered detectors inactive during the corresponding time period.

The STFs are then dispatched to the Event Processing Nodes (EPNs) for aggregation, as shown in Fig. 5.2. The STFs related to the same time period and from all FLPs are received by the same EPN and aggregated into a TF. The EPNs provide information about their capability to receive more data in the immediate future. This information is used by the load balancing system to prepare a list of candidate EPNs ready to receive further STFs. This process implements a mechanism for smooth EPN load balancing and makes it possible at run-time to adapt the data flow to the capabilities of the EPNs and to temporarily bypass EPNs which are overloaded. Moreover, if an EPN breaks down or does not publish the information about its availability, the load balancing simply removes this EPN from the list of possible destinations for the STFs, avoiding possible loss of data or system hang-up.

The load balancing system is also used to regulate the dataflow from continuously read out detectors. In the case of data saturation of the whole system, none of the EPNs will declare their readiness to receive data. If this happens, data received are then discarded by the FLPs. This mechanism will create dead time in a way similar to the busy with a traditional trigger system.

EPNs perform the reconstruction for each detector, further reducing the data by an average factor of 8 (see Chap. 9). The fully compressed TFs are then stored on disks in the O^2 facility or Tier 0 / Tier 1 data centres. Permanent storage is used for archiving.

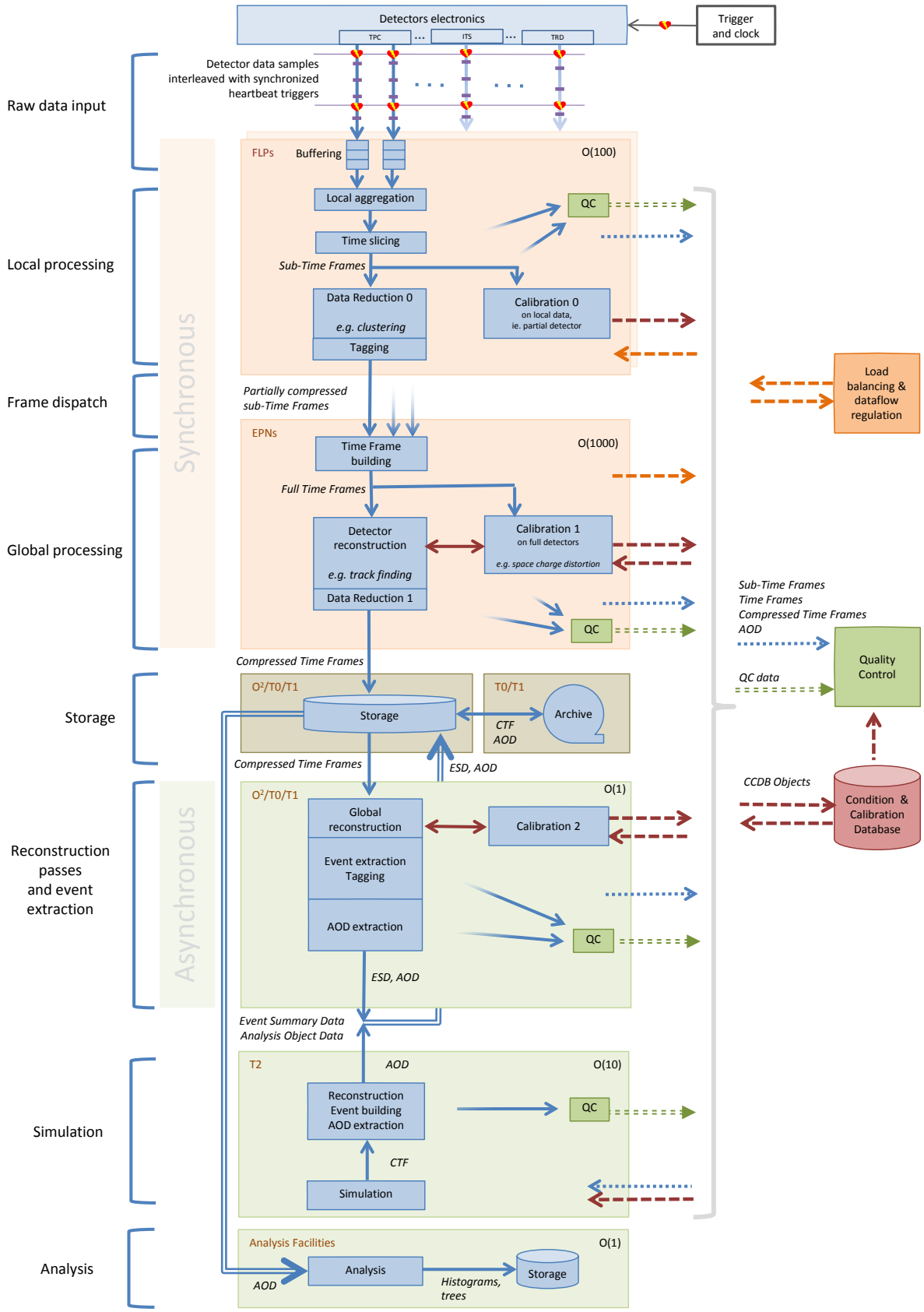

Figure 5.1: Data flow and processing pipeline of the O^2 system.

Global reconstruction and events extraction will be performed asynchronously in iterative passes, either on the EPNs, or on the Grid to offload processing. The ESDs produced are stored at the O^2 facility. The AODs are transferred to dedicated facilities for analysis.

Calibration is integrated at different processing stages on the FLPs and EPNs, both to minimise data transport and to make results available as early as possible in the chain; this is the key to real-time data reduction. The Condition and Calibration Data Base (CCDB) is populated and used at all stages, synchronously and asynchronously.

The global data flow also includes data sampling at all levels, in order to feed the Quality Control (QC) system as needed; the results will be stored in a dedicated database.

Simulation is done in Tier 2 centres, and resulting AODs and the physics data are analysed in the Analysis Facilities.

Figure 5.3 shows the hardware pipeline with the estimated number of processing nodes and throughput at the different online stages. As detailed in Chap. 10, of the order of 250 FLPs and 1500 EPNs will be needed to cope with the data processing. The global detector read-out rate is 1.1 TB/s over 8300 read-out links. Each FLP produces up to 2.5 GB/s after data reduction for a total of 500 GB/s. The TFs built at 50 Hz by the EPNs are 10 GB each before compression. The EPNs further reduce the data to a total peak throughput to storage of up to 90 GB/s which makes a local write throughput to disk of 60 MB/s per EPN after compression. One of the key components is the network fabric switching the data from FLPs to EPNs at 500 GB/s or 4 Tb/s.

5.2 Data flow

5.2.1 Data model

A sustainable reduction factor of about 14 in the data throughput can only be achieved by running at least the initial calibration and reconstruction steps as pipeline processes, synchronous with the data acquisition. The read-out scheme of the ALICE detector calls for a hierarchical organisation of the data model to follow the data flow from individual data links to EPNs processing complete TFs. The data model throughout the O^2 facility is designed to assemble the data in a fast and optimal manner for processing by the reconstruction algorithms while taking into consideration navigation and performance issues. The following paragraphs address the data model starting from the main requirements as presented in Chap. 3. The scope is limited to the "transient" form of data within the processing pipeline up to the stage of writing it in the first persistent format.

The data flow and processing in Runs 3 and 4 will be centred on Time Frames, which size is justified in 5.2.2; all data blocks will need to have clear time identifiers. The data model does not require a strict format specification for the data blocks but it has to impose format constraints for the metadata (headers) describing them. This is mainly to allow seamless implementation of interfaces to identify and group different data types, which is an important requirement in later processing stages.

Fast navigation among the different data blocks produced within a given TF is another critical requirement of the data format. The data corresponding to different but matching TFs will lie on different processing nodes. For this reason, there is a strong need for self-containment of data, not only to allow TFs to be regrouped for event processing purposes, but also to perform opportunistic processing when resources are available. This uniformity is an important principle in the data model design: the data blocks on FLP nodes should look the same on EPNs so that processing tasks can be interchangeable.

Data persistency is a critical issue. During the synchronous phase the raw data cannot be made persistent due to the high data throughput. Therefore the data will only be made persistent after the replacement of raw data by the more compact result of the synchronous calibration and reconstruction. The subsequent

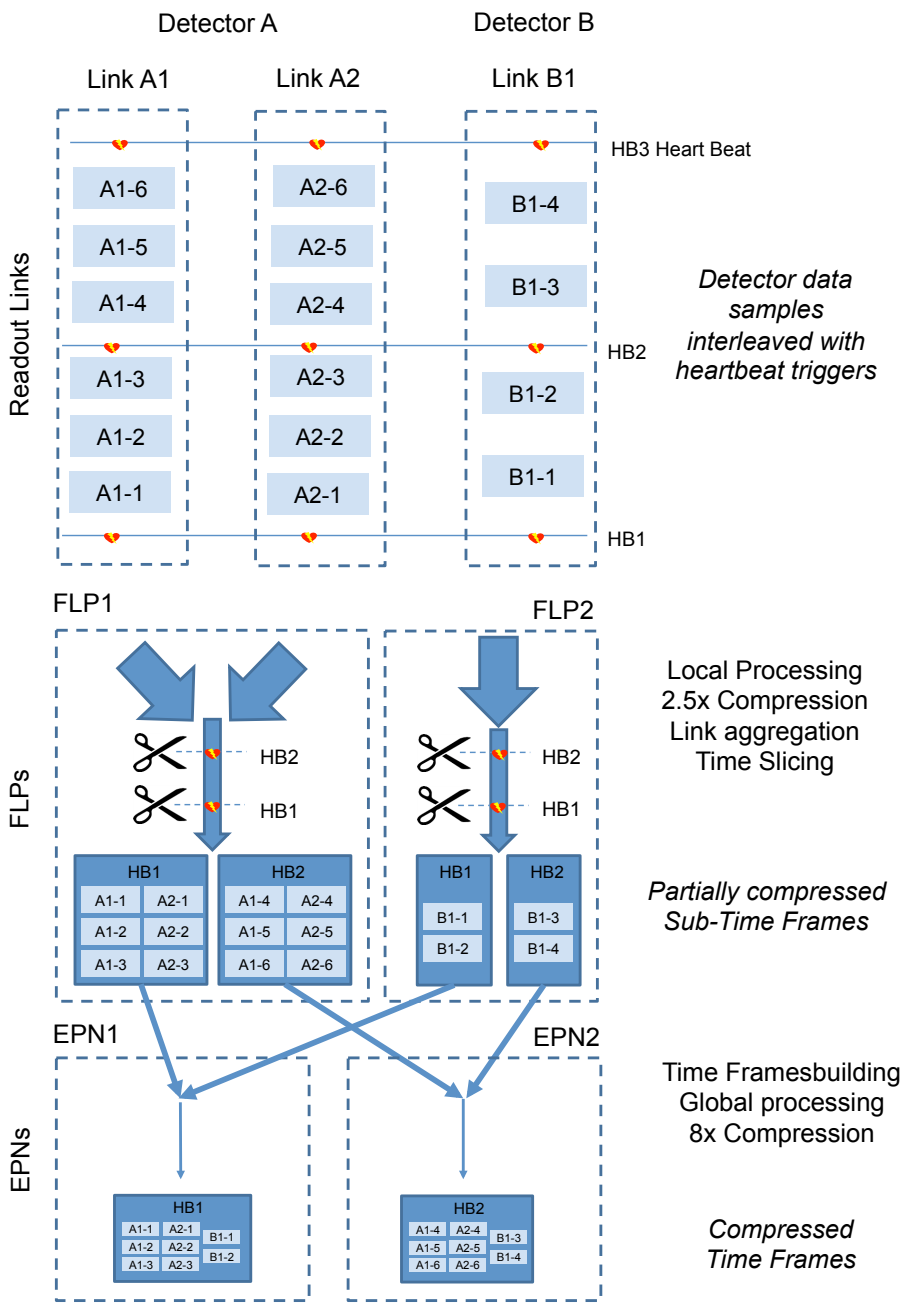

Figure 5.2: Aggregation of data, from the fragments read-out from the detectors to the full Time Frames built on the EPNs.

asynchronous passes of reconstruction, calibration or analysis will produce persistent formats representing more elaborated data. Consequently, the main requirements for the persistent format are the support for schema evolution and binary object format.

5.2.2 Time Frame size

The choice of the TF size (i.e. the duration of the time window or triggers period of the HB) depends on several criteria.

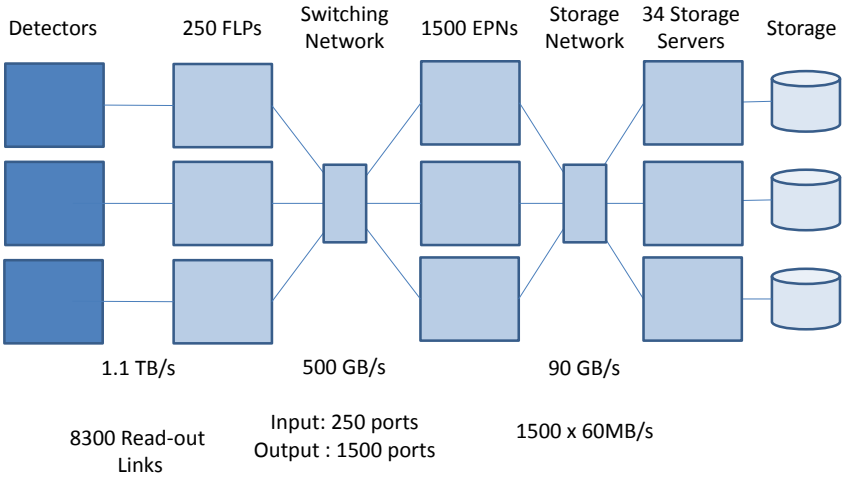

Figure 5.3: Possible hardware implementation of the logical data flow described in Fig. 5.1.

– A fraction of the data at the TF edge can not be used due to the TPC drift. The fraction lost is:

$$0.1/t$$

where t is the duration of the Time Frame in milliseconds. For example, a TF spanning 100 ms would cause a data loss of 0.1%. Having longer TF is better in this respect.

– The TFs are distributed over the entire EPN farm. It would be preferable to keep the interval between two consecutive TFs received by a given EPN on a few minutes level, in order to minimize discontinuity in the calibrations at the synchronous stage. A TF of 0.1 s in a farm of 1500 EPNs would lead to a cycle time less than 3 minutes.

– The calibration data produced within each single TF will have some fixed size overhead, since a lot of data will be accumulated in fixed size containers (histograms). Therefore, shorter TF would mean more frequent input to CDB and also larger data rate.

– The TF should be a multiple of the shortest calibration update period, currently estimated at 5 ms for TPC Space Charge Distorsion (SCD) maps.

– Longer TFs reduce the overhead query of some of the calibration parameters, as some calibration data from each side of the TF are needed to interpolate values with greater precision. (e.g. take an extra 5ms SCD bin on each side of the TF). However, the overall query rate is still small compared to the TF data (around 20 MB of calibration data per second of physics data, i.e. less than 0.1%).

– Shorter TFs reduce the size of data to be handled on every EPN, and ease staggered data transfer from the FLPs. On the EPNs, we anticipate the need of buffers for 3 full time frames (one receiving, one processing, and one sending).

Any TF between 20 ms and 100 ms is considered to be adequate for calibration/reconstruction. Having a finer TF granularity makes it easier to distribute and buffer the data; the selected design value used for

the TF duration is 20 ms, or a TF rate of 50 Hz. This results in a TF size of 10 GB on the EPN (before compression). The corresponding 0.5% of data unusable at the frame boundaries is still acceptable, and buffers sizes of a few tens of GB can easily be implemented. For a farm of 1500 EPNs, each EPN would receive a new TF every 30 s. Finally, a TF corresponds to 1000 interactions in normal running conditions (Pb–Pb at 50 kHz), which is enough to compute its calibration parameters.

5.2.3 Read-out links

The data are transferred from the detectors to the O^2 system via optical read-out links of three different types. The detectors installed or upgraded during the LS2 will use the GBT. The GBT link is a bi-directional link multiplexing control, data and trigger information. DDLs 1 and 2 links will be used by the detectors which do not modify their electronics during LS2. The DDL link is bi-directional: one direction is open for data taking, receiving data from the detector. The other direction is used to send slow control information or download configuration data. Several read-out links are multiplexed into the FLP I/O bus by dedicated adapters.

5.2.4 FLP and EPN

The O^2 farm will consist of two different categories of computing nodes, corresponding to the two data aggregation steps.

The FLPs will collect the detector data at a rate of 1.1 TB/s (see Tab. 3.1) over approximatively 8300 read-out links (see Tab. 3.2). Each FLP will get the data from up to 48 optical links at up to 3.2 Gb/s. The data from these streams will be compressed by a factor of 2.5, merged, split into Time Frames and buffered until they are sent to the EPN. The most heavily loaded FLPs are the ones of the TPC inner chambers with an aggregate sustained average of 50 Gb/s at 50 kHz interaction rate. The FLPs will therefore need an I/O capacity of the order of 100 Gb/s as peak input, 50 Gb/s as sustained input, and 20 Gb/s as sustained output.

The EPNs will perform the second step of data processing: assigning each cluster to a track.

This step will provide an additional reduction factor of 8 in the data volume. The total throughput to data storage reaches a peak of 90 GB/s which makes a local write throughput to disk of 60 MB/s per EPN after compression.

The characteristics of the FLPs and EPNs are summarised in Tab. 5.1 which shows the peak and average needs. For the FLPs, the peak needs correspond to a period when all read-out links would use their maximum bandwidth. For the EPNs, the peak need corresponds to the period at the beginning of the fill when the luminosity and therefore the data throughput reach their maximum value.

Table 5.1: Hardware characteristics for FLP and EPN nodes.

Type	Number of Nodes	Input bandwidth		Output bandwidth	
		Peak (Gb/s)	Average (Gb/s)	Peak (Gb/s)	Average (Gb/s)
FLP	250	100	50	40	20
EPN	1500	10	2.7	0.48	0.33

5.2.5 The FLP-EPN network

As shown in Fig. 5.3, the network fabric that connects the FLPs to the EPNs must be capable of sustaining high throughput traffic (500 GB/s or 4 Tb/s) to transfer, for each TF, the corresponding STFs from all FLPs into the same EPN for aggregation and processing.

A relatively high number of point-to-point transfers must be done in parallel to keep up with the total traffic. At the same time, the data traffic and processing load should be balanced and distributed over the entire EPN online farm. This “fan in”-“fan out” traffic shape requires the core switching fabric to be able to cross-switch the entire traffic without blockages.

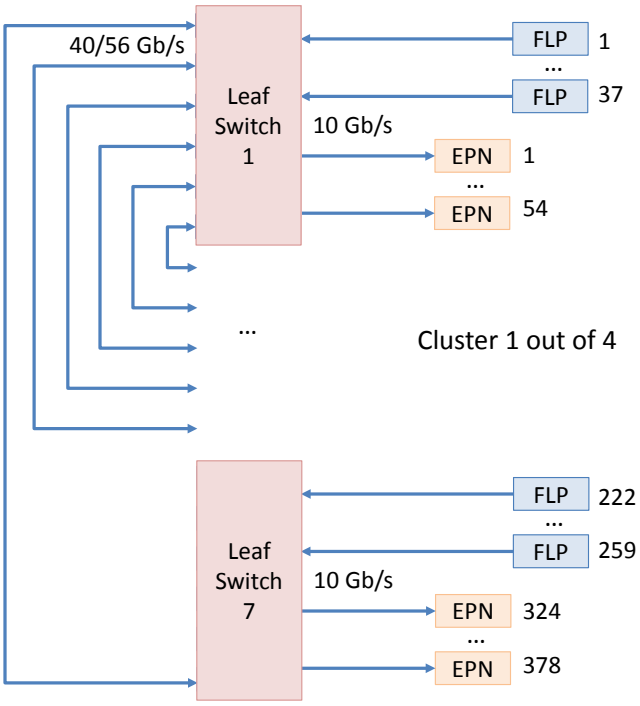

Figure 5.4: FLP-EPN network layout with four EPN subfarms.

Three possible layouts are considered:

- The first one consists of connecting all the FLPs and EPNs to a single large switching network with all ports having the same bandwidth sufficient for the nodes with the highest needs as shown in Fig. 5.3.

- The second layout consists of adapting the port bandwidth to the estimated need of each type of node in order to reduce the cost. This option is justified by the large difference between the peak output bandwidth of the FLP (40 Gb/s) and the peak input bandwidth of the EPN (10 Gb/s). This second layout shown in Fig. 5.4 has also the characteristic of dividing the EPN farm into four identical parts and therefore requiring four smaller networks. This layout is also practical for a staged deployment. Each FLP is connected to each of the subfarms. This layout reduces the network cost by splitting the total data traffic into four. The data flow must take into account the actual topology of the network at the application-level. The application should apply some sort of traffic shaping to avoid the saturation of any link or switch and to monitor performance to ensure there is adequate load balancing between the EPNs.

- The third layout consists of splitting the fan-in and fan-out functions into two separate layers of the network as shown in Fig. 5.5. The fan-in function is performed by a standard switch or a combination of switches and aims at minimising the number of high-speed links used to perform the assembly of Time Frames. The fan-out is performed by new types of nodes, the super-EPN (SEPN), in charge of assembling complete Time Frames and distributing them to EPNs.

A further optimization of this third layout being studied and simulated now consists of combining the SEPN and FLP functions in the same node. This would reduce the number of nodes on the high performance network while having more nodes performing the SEPN function.

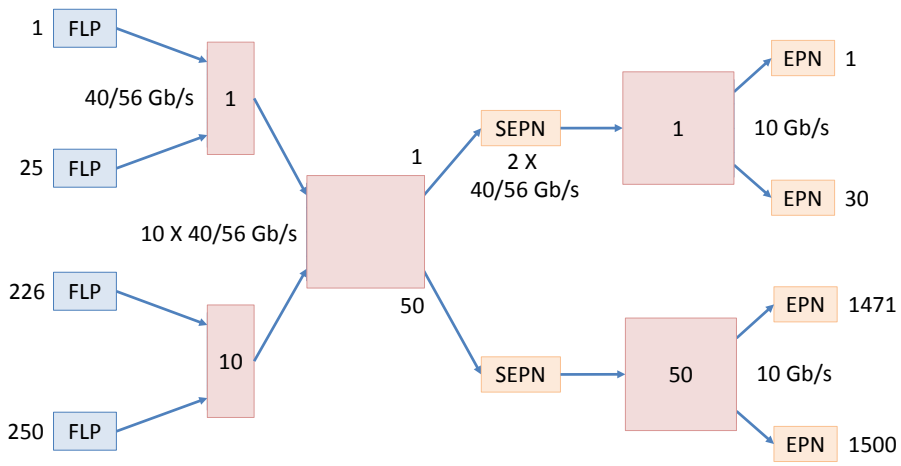

Figure 5.5: FLP-EPN network layout with SEPNs in charge of assembling and distributing the TFs to the EPNs.

The decisions about the actual network layout implemented and the technology used will be taken later because the evolution of the different technologies will have a significant impact on the most cost effective solution.

5.3 Data storage

The O^2 architecture includes a storage system used by the EPN as a buffer between the synchronous and asynchronous steps of the data processing. The storage system is also used for interfacing the O^2 system with the Grid: temporary data parking and permanent archiving in Tier 0; data processing in Tier 0 and Tier 1s; simulation in Tier 2s. The data transfer operations between the O^2 facility and the Grid are performed by dedicated nodes, of the order of 10 Data Movers (DM), in order to avoid overloading the EPN nodes already saturated by data acquisition and processing tasks. The storage back-end will be physically decoupled from the EPN nodes but still provide a global storage space accessible by all EPNs and DMs. Sufficient bandwidth has to be guaranteed to avoid data pile-ups in the EPNs and DMs.

5.3.1 Condition and calibration database

As part of the data storage, a CCDB is foreseen. This will be the database used to store the information related to the detectors calibrations, needed mainly for reconstruction and simulation purposes. The database will be implemented on the basis of the experience with the Offline Condition DataBase (OCDB) in Runs 1 and 2.

The design of the database will offer simple read/write/update/delete access patterns, based on a simple access configuration (e.g. using strings), but the usage will limit the number of non-read operations. The database will be constructed following the concept of "validity" which will guarantee that the desired object is returned depending, for example, on whether the query was aiming at the latest version of the object, or at some different criteria. In this respect, the versioning of the objects stored will be a key feature of the CCDB.

The technical implementation of the CCDB will allow low latency when retrieving an object, scalability (fundamental due to the expected growing size of the database and to the high number of processes which

are foreseen to connect to it), and will include a publish/subscribe mechanism in order to keep up to date all the services requiring access to the CCDB. A redundancy policy will be established, in order to avoid loss of data.

Typically, the content of the objects stored in the CCDB will be time dependent essentially because the data processing in Runs 3 and 4 will be based on time frames. Metadata will be associated to the objects in order to describe their main properties, such as the data taking period to which they refer, the calibration type, the source etc.

5.4 Calibration, reconstruction and data reduction

There are two phases of reconstruction and calibration, synchronous and asynchronous:

- Synchronous: performed during the data taking.
- Asynchronous: decoupled from the data taking process. Its aim is to provide analysis grade data for the full detector, analysable in terms of individual collisions.

The goal of online reconstruction and calibration is to reduce the data volume as much as possible while remaining compatible with the required physics performance.

5.4.1 Calibration and reconstruction flow

Figure 5.6 shows the calibration and reconstruction flow divided in five conceptual steps which will be described in more detail in Sec. 8.1.

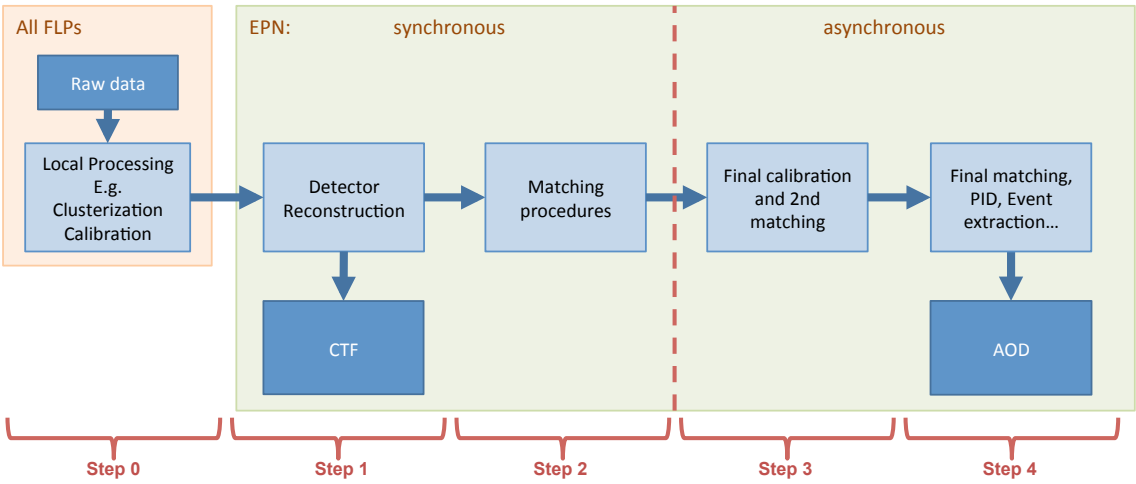

Figure 5.6: Schematic outline of the reconstruction and calibration data flow.

The output of the FLP processing is merged and shaped into the STF format and shipped to EPNs for integration into TF and to carry out the track and event-level reconstruction. The principal difference of Runs 3 and 4 compared to the current operation of ALICE is the presence of continuous read-out detectors like TPC and ITS. For this reason, and because a given EPN does not receive adjacent TFs, the collisions corresponding to the edges of the TF will be discarded; part of the collision data may appear in a different TF on a different EPN. The length of the time interval to be discarded is $\sim$100 μs corresponding to the full drift time in the TPC. Reconstruction and calibration on the EPNs refers to

single TF processing. As it is an essential part of the TPC data volume reduction schema, only the TPC full track finding described in Sec. 8.1.3 must be done synchronously.

The calibrations that are produced on EPNs during the synchronous stages of processing will be available during the same stage only for the data that will come to the same EPNs; no cross-talk is foreseen between different EPNs. This is considered possible if no calibrations requiring a large amount of the statistics is needed. The drawback of such an approach is the case when calibrations obtained from standalone reconstructions on the FLPs are needed. The first portion of data on each EPN will not be calibrated initially and a second processing will be necessary. Nevertheless, EPNs should be able to read from the calibration database in order to make use of the calibrations previously produced. The details of the calibration, reconstruction, data reduction and event extraction procedures will be presented in Sec. 8.1.

5.4.2 Data volume reduction

Reduction of the raw data volume shipped by detectors will start already on the FLPs, by converting raw to clusterized data and compressing by lossless algorithms. It will be followed by the lossy data reduction during the synchronous stage of the reconstruction on the EPNs, where as much as possible clusters which have no chance to contribute to physics analyses (noise, background from δ-rays and beam-gas collisions) will be rejected. An additional factor in data compression will be gained by exploring the correlations between the clusters and the tracks. They can be attached to during the synchronous reconstruction stage. Details will be presented in Chap. 9.

5.5 Data quality control and assessment

The aim of data QC is to provide a prompt online feedback on the quality of the data being recorded and on the processes underlying the handling and transformation of this data, such as reconstruction and calibration. It is also of great importance to control the quality of subsequent reconstruction/calibration steps performed asynchronously to the data taking. As a consequence, the QC covers what is known as Quality Assurance (QA) and Data Quality Monitoring. The QC chain's output should consist of an assessment of the data quality, usually computed automatically. The QC decision is used to check the transition from uncompressed to fully compressed data.

The Event Display allows experts and shifters to visually check the events in a qualitative, non-statistical way. For this reason, it is considered a part of the QC. It is designed to visualise, manipulate and analyse physics events in a three dimensional environment. The system will be able to receive events from different sources at different stages of the reconstruction chain and with different formats such as Time Frames, CTF, raw data or AODs. The Event Display will provide features such as bookmarks, history browsing and events filtering.

5.5.1 General QC workflow

The QC system can be described by a workflow as in Fig. 5.7. Data from the online data flow or from data storage are processed by the QC. User-defined algorithms are applied in order to generate and populate the monitoring objects such as histograms. The quality assessment of these objects comes next, either at the same time or later in a separate step. Finally, the monitoring objects as well as the degree of their quality are stored and made available for visualisation, post-processing and for use directly from the data flow (not shown in the figure). One of the most important aspects of post-processing is the capacity to trend the monitoring objects and results over time.

5.5.2 Detector requirements

The detectors have provided the estimated number of monitoring objects they plan to produce at each stage as well as the absolute or relative quantity of data to be monitored to have relevant QA.

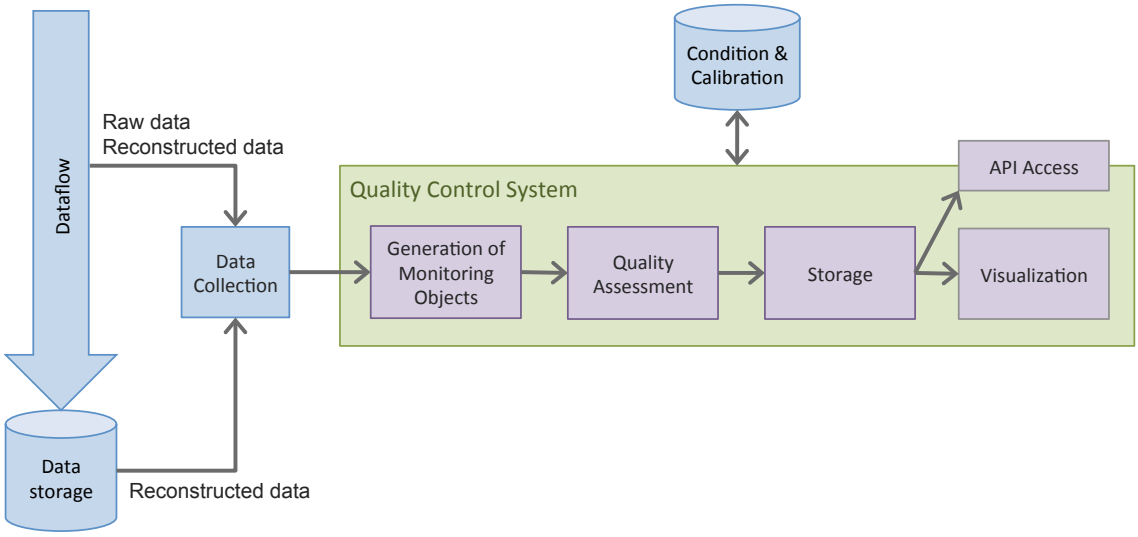

Figure 5.7: Quality control and assessment general workflow.

The total number of monitoring objects foreseen to be produced concurrently is of the order of 2000. Experience from Run 1 shows that it is safer to aim at a higher number of monitoring objects to take into account the natural tendency to produce more QA objects than initially planned. In view of this, the baseline for the QA system architecture and design is ∼5000 objects with peaks at ∼10000.

While most detectors only need to monitor 1-10% of the raw data or TFs, a few others need to run on 100%. Some of these tasks will need a large amount of computing power.

5.5.3 Input and output

The QC system receives data at different stages of the online data flow or from the offline data storage. Information about the conditions under which these data were collected and the associated calibration, is also made available to this system. The QC provides objects (e.g. histograms, graphs or values) and a quality assessment for nearly all of them. The quality assessment is provided by the setting of different quality flags: Undefined; Ignored; Good; Bad; Warning. In order to assess or refine the quality of an object, different types of monitoring can be carried out, depending on the time and information available. In addition, the QC output objects will require metadata to index the monitoring information and reduce the need for collecting such information from other systems.

5.5.4 Automation

In general, the QC results cannot be analysed manually. The main reason for this is that the latency introduced by non-automatic (i.e. human) checks is too high to allow efficient feedback and remedial action if something goes wrong. Automatic checks will categorise data as good or bad so that the experiment and the O^2 systems can react accordingly. The framework will provide a set of common tools to easily check the quality of the monitoring data, for example, assessing whether the mean value of an histogram is below a certain limit.

Automatic checks on QC objects are important for taking early decisions on the data flow, especially in the case of critical parameters which have reference values or known behaviour parameters. QC objects, such as histograms, are evaluated by algorithms that produce a clear and discrete value, which can be based on references representing its level of quality. Such values are important for helping shifters to make evaluations of the data quality and to make decisions on the continuation of data taking. In addition, they inform the data flow procedure so that automatic and unambiguous decisions can be made.

The automatic evaluation of QC objects is important for critical online processing stages like calibration and reconstruction. At a later stage, it will be used for deciding on the quality of subsequent calibration and reconstruction passes and whether a data taking period is of sufficient quality for physics analysis.

5.5.5 Access to QC results

QC output is available to Collaboration members worldwide under the form of downloadable and manipulable object files and interactive web-based items such as histograms.

5.6 Facility control, configuration and monitoring

5.6.1 Overview

The Control, Configuration and Monitoring (CCM) components of the O^2 system act as a tightly-coupled entity with the role of supporting and automating day-to-day operations. The Control system is responsible for coordinating all the O^2 processes according to system status and monitoring data. The Configuration system ensures that both the application and environmental parameters are properly set. Finally, the Monitoring system gathers information from the O^2 system with the aim of identifying unusual patterns and raising alarms. Figure 5.8 shows the relationship between the CCM components.

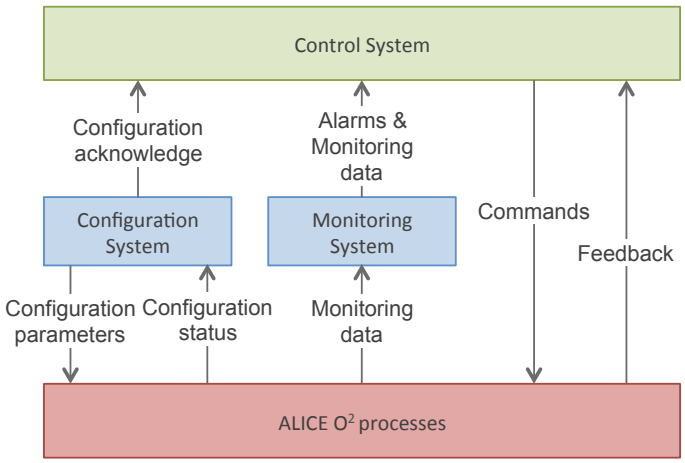

Figure 5.8: Overview of relationship between CCM systems.

For the global operation of ALICE, the CCM systems interface with the Trigger and DCS systems to send commands, transmit configuration parameters and receive status and monitoring data (as shown in Fig. 5.9). The CCM systems also interface with the LHC to automate certain operations and keep a record of data taking conditions.

The asynchronous data processing is initiated by the CCM systems whenever enough computing resources are available in the O^2 facility and/or the Grid. To offload asynchronous data processing tasks to the Grid, the CCM systems dispatch them as normal Grid jobs. This approach has the advantage of reusing the existing and proven Grid middleware, thus minimising O^2 software development.

The use of the O^2 farm as a Grid facility is controlled by the CCM systems, depending on the availability of resources. The CCM systems only enable/disable the access to the resources, while the actual handling of the Grid tasks are performed by the Grid middleware.

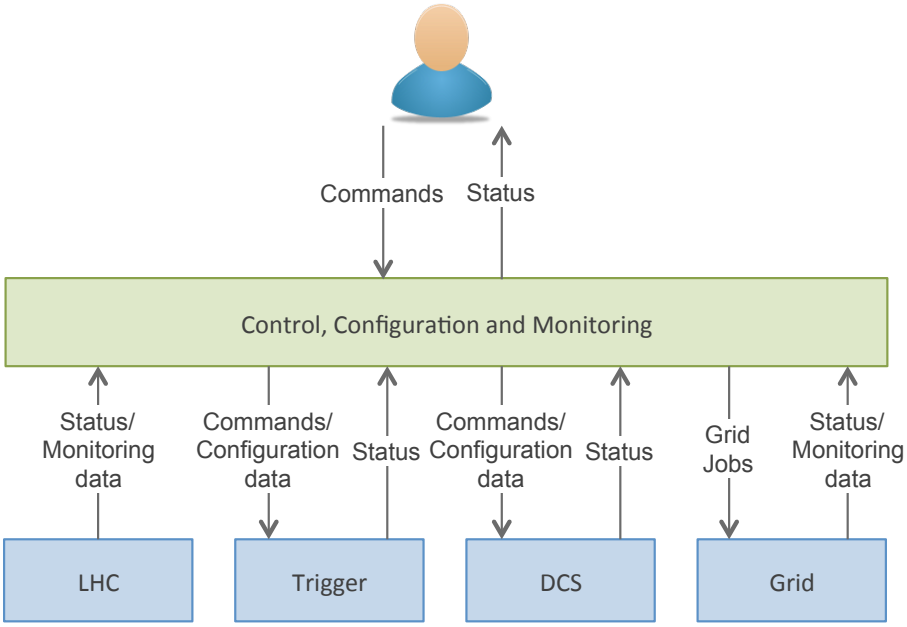

Figure 5.9: CCM interfaces with external systems.

5.6.2 System functions

Control

The Control system starts and stops the processes running on the O^2 facility. This includes not only the processes implementing the different functional blocks (data read-out, data reduction, reconstruction, etc.) but also processes providing auxiliary services such as databases and the Domain Name Server (DNS). The Control system also sends commands to running processes according to changing conditions and operational requirements. Typical commands include pausing or resuming of ongoing actions, applying a new configuration or terminating the current action and moving to a standby status. Processes must acknowledge when they receive a command and inform the Control system when the command execution is completed.

The Control system executes and coordinates the different O^2 functions. These functions are defined as sequences of actions known as ”Tasks”.

The Control system reacts to internal and external events to achieve a high level of automation thereby reducing the need for human intervention in the experiment’s operations. Identified external events include LHC state changes (stable beams, beam dump) and ALICE detector status. Internal events include: process failures; system load greater than predefined threshold; non-nominal data quality. Starting and/or stopping data taking activities based on LHC and detector status are examples of automatic Control system actions.

Configuration

The Configuration system distributes the configuration from a central repository to all processes in the O^2 system. The system supports both static configuration where a process is restarted in order to read the new configuration parameters and dynamic configuration where the parameters are loaded on-the-fly, without stopping and restarting the process.

The Configuration system takes care of software installation and configuration, including base Operating System, software packages and configuration files.

Monitoring

All O^2 components are capable of providing monitoring parameters, like heartbeats, processing status and other critical metrics. All these data are collected by the Monitoring system where they are processed in quasi real time in order to trigger alerts or take automatic corrective actions. This system also aggregates monitoring data streams and persistently stores the relevant metrics to provide high-level views of the system and to support analysis over long periods.

The Control system can assess the health of the system in general and trigger actions accordingly, for example if a process fails to report as running or critical services report an error condition.

Logging

Log messages are considered to be part of the monitoring data and therefore use the same infrastructure. Dedicated visualisation tools allow the shift crew and remote experts to display, filter and query log messages, thus providing a global monitoring perspective of all O^2 components.

5.7 Detector Control System

The DCS ensures safe, reliable, and uninterrupted operation of the experiment. It also serves as an important communication exchange point, providing vital data for detector operation, physics analysis, and safety systems as well as for external services, including the LHC. The upgrade of the online and offline computing into the O^2 system will modify some of the interfaces of the DCS as explained below.

The interfaces of the DCS are shown in Fig. 5.10. The DCS data are processed in the Central Supervisory

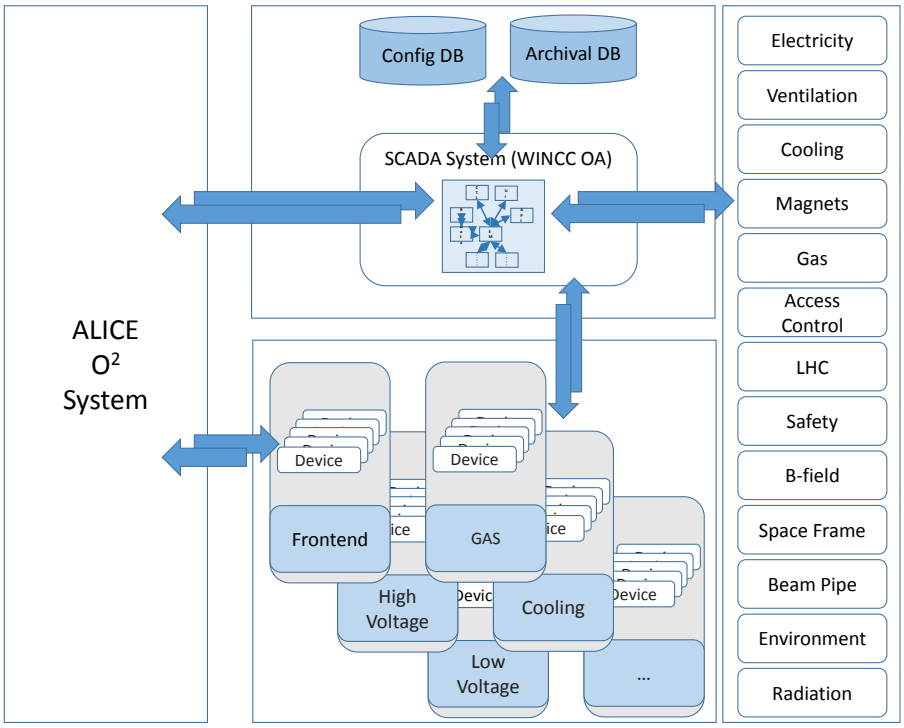

Figure 5.10: DCS interfaces with detector devices, external services and the O^2 system.

Control and Data Acquisition (SCADA) system, based on WINCC Open Architecture (WINCC OA), provided by SIEMENS. The core of the control system is autonomous and serves its purpose even in the absence of external systems and services (i.e. during network outage or external system maintenance).

All devices are continuously monitored by WINCC OA and acquired values are compared to predefined thresholds. In case of significant deviations from nominal settings, the SCADA system can take automatic remedial action, or alert the operator. The detectors added or upgraded during the LS2 will make a massive use of GBT-based read-out links. These links are interfaced to the O^2 system and are used for transferring both physics and control data. The electronics of these detectors will therefore be accessed by the DCS through the O^2 system.

The operational limits, device settings and configuration parameters are stored in the Configuration database or directly in the WINCC OA systems. Whenever the experiment mode changes, for example from standby to data taking, the SCADA system reads the new settings from the database and reconfigures the control systems and devices accordingly. All parameters tagged for archival are sent to the archival database (ORACLE). The stored data are available for later retrieval either directly from WINCC OA or by external clients. The detector conditions data as well as parameters acquired from external systems are transmitted to the O^2 farm at regular time slots via a dedicated FLP. The transmitted data frames contain the full map of all monitored parameters. These conditions data are required by the O^2 system for the online reconstruction.

The DCS interacts with external systems and services such as cooling, safety and gas. The acquired information is distributed to detectors and ALICE systems and feedback is provided back to the services. The synchronisation between the DCS and O^2 components is achieved using the Finite-State Machine (FSM) mechanism.

Chapter 6

Technology survey

6.1 Introduction

This chapter gives an overview of available computer hardware and software. Performance characteristics, costs and efficiency are investigated and criteria defined for the selection of appropriate computing platform for the future O^2 compute facility. A considerable part of this study is devoted to the identification and development of benchmarks that can be used to determine the applicability of a system for typical workloads.

For each technology, the parameters or the devices relevant for the O^2 project are indicated.

6.2 Computing platforms

Short and mid-term heterogeneous target architectures with power-efficient, highly parallel performance have been considered. Requirements are indicated for highly portable languages and frameworks that will provide flexibility for future hardware choices.

6.2.1 Input-Output (I/O)

The PCI-Express (PCIe) bus is the state of the art I/O bus in recent server PCs [1]. Almost any interface to the outside, like network interfaces, graphic accelerators and even storage devices can be connected via PCIe. The standard has been available for several years and has evolved continuously. Each new generation has come with increased bandwidth capabilities while retaining backward compatibility down to the very first release of the specification. In the last few years, the PCIe root port has been moved from near-CPU I/O-Hubs (Nehalem architecture) into the CPUs themselves (SandyBridge / IvyBridge architecture), boosting the performance of the bus. An overview of the evolution of the PCIe standard is shown in Tab. 6.1. The first generation of PCIe Gen1 uses a link rate of 2.5 Gb/s per lane. This corresponds to a theoretical raw throughput of about 250 MB/s per lane. The PCIe Gen2 link rate as well as the throughput per lane has been doubled. PCIe Gen3 progress comes mostly from changes to the serial encoding of the data allowing a further doubling of the bandwidth, and to a lesser extend from the link rate increased only by a factor of 1.6. Typical devices of PCIe Gen2 or Gen3 have 8 or 16 parallel lanes. The bandwidth of the bus that is actually usable for payload transfer is about 70-80% of the raw bandwidth.

At the time of writing, standard server and desktop PCs already come with PCIe Gen3. The specification for this generation was announced in 2007 and finally released in 2010. Typically, the industry waits a few months for the first devices with the new generation bus to be released and another few months to resolve initial bugs or incompatibilities. The next Generation, Gen4 of the PCIe standard has already

Table 6.1: Evolution of the PCI Express standard [1].

PCIe Generation		Gen1	Gen2	Gen3	Gen4
Specification Announcement		2002	2005	2007	2011
Specification Release		2003	2006	2010	late 2015
First Products		2003	2007	2012	
Per Lane Bitrate	(Gb/s)	2.5	5.0	8.0	16.0
Encoding		8b10b	8b10b	128b130b	
Per Lane Raw Throughput	(MB/s)	250	500	985	1969
x16 Raw Throughput	(GB/s)	4.0	8.0	15.75	31.51
x16 Payload Throughput	(GB/s)	∼3.5	∼7.0	∼13.5	

been announced and its specification release is expected for late 2015 [2]. Extrapolating from past experience, Gen4 devices should be usable in production environments in 2016/2017.

The throughput of PCIe Gen2 and Gen3 links has been extensively measured during the development of the Run 2 read-out board (C-RORC), the evaluations of possible future read-out boards and the characterisation of GPUs. The plot on Fig. 6.1 shows measurements of a dual-socket IvyBridge machine with two C-RORCs (PCIe Gen2 x8) and one GPU (PCIe Gen3 x16). Even when enabling high speed I/O transactions with the GPU while both C-RORCs are active, the performance of any of the devices was not affected. In this test, a combined throughput of ∼17 GB/s distributed over 3 PCIe devices was measured in the system.

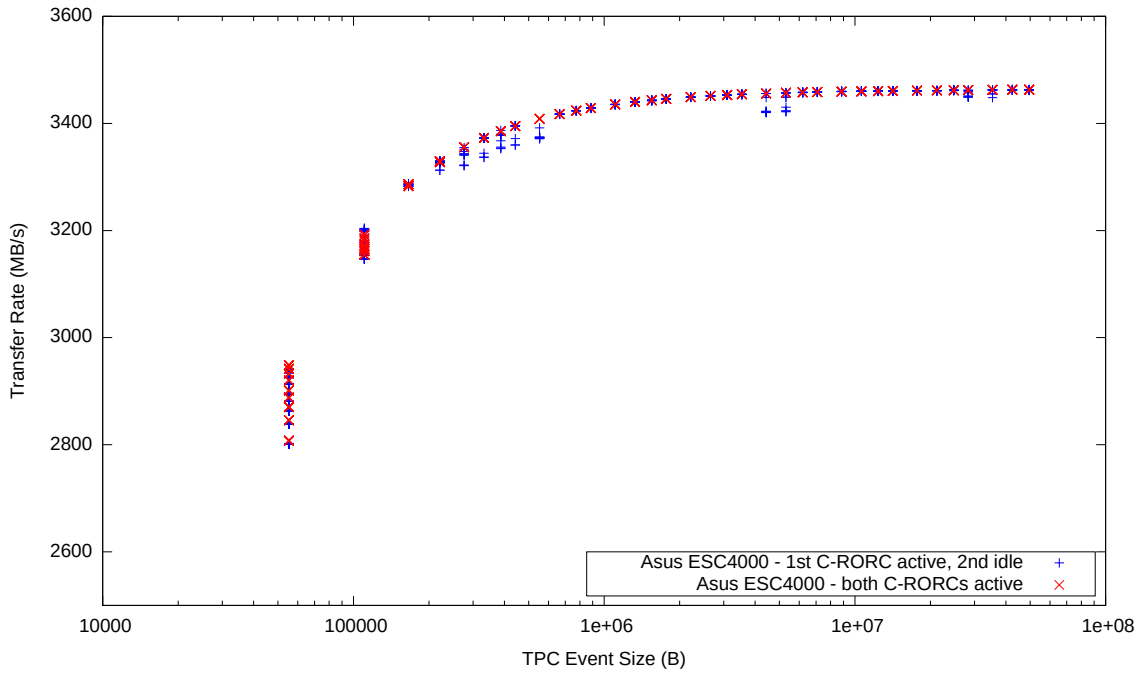

Figure 6.1: Direct Memory Access (DMA) to host performance for multi-device PCIe operation using two C-RORCs and a GPU. The blue + show the DMA throughput performance of a single C-RORC board, while the second C-RORC and the GPU were idle. The red x show the same measurement with also the second C-RORC transmitting data via DMA at its full speed.

The plot on Fig. 6.2 shows the throughput measurement of a Xilinx Virtex-7 XC7VX330T FPGA board with a 8 lane PCIe Gen3 interface using the Xilinx example design for PCIe DMA on a Supermicro X9SRE-F machine. This design already provides a throughput of 5 – 6 GB/s or 40 – 48 Gb/s both from

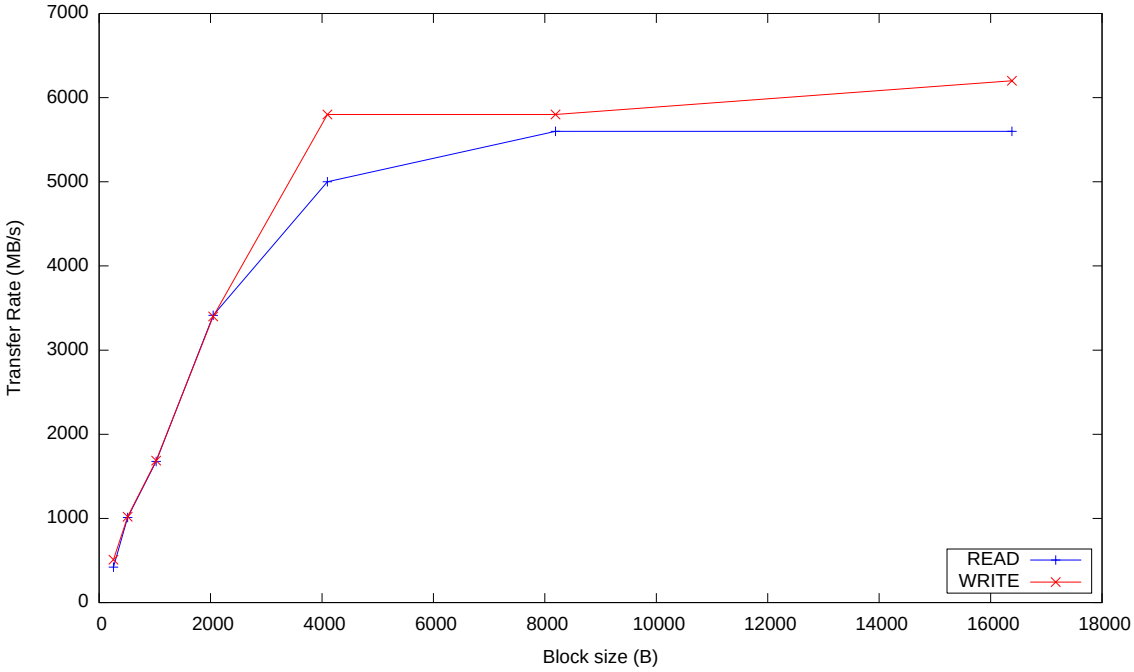

Figure 6.2: DMA throughput measurements on a Xilinx Virtex-7 FPGA with a PCIe Gen3 x8 interface.

FPGA to host and from host to FPGA.

The expected input/output data rates on the FLP and EPN nodes are in the capability range of PCIe Gen3. The most demanding O^2 application will be the FLPs with two CRUs interfacing 24 GBTs at 3.2 Gb/s for a total of 77 Gb/s or 9.6 GB/s. Today's bus technology can deliver 80 Gb/s on a 16 lane PCIe Gen3 and is therefore sufficient for the development of the O^2 system without relying on future developments of PCIe generations. The backward compatibility of future PCIe generations will still permit the use of hardware developed and tested for earlier generations, making possible the use of accelerator cards like NVIDIA Kepler, the Intel Xeon Phi, the AMD FirePro and FPGA based processing boards in the O^2 system.

6.2.2 Memory bandwidth

DDRx memory is the standard for host memory in current computers. Its maximum throughput over the last years has continuously progressed as shown in Fig. 6.3. Current Intel Ivy-Bridge CPUs support DDR3 at 1866 MHz achieving about 15 GB/s per memory channel. The aggregate of this with four channels per processor is 60 GB/s for a total of 120 GB/s for a standard dual-socket server. After the integration of the memory controllers in the CPU, the first processor families showed significant NUMA (Non Uniform Memory Architecture) effects, if a CPU core accessed memory connected by the other CPU socket. These effects are significantly attenuated with Intel Ivy-Bridge CPUs, and since event-based parallelism can be used in ALICE, local memory will suffice in most situations. NUMA benchmarks have been developed to measure these effects and will be the object of continued observation.

The most demanding application will again be the FLPs with two CRUs. It requires the transfer of raw data at up to 9.6 GB/s in and out of memory and the transfer of compressed data at at up to 3.8 GB/s in and out of memory for a total of 26.8 GB/s to and from the memory of one CPU. This requirement is compatible with the performance of up to 60 GB/s achievable with the present DDR3 memory. The baseline FLP will therefore consist of a server with two CPUs, each using one PCIe Gen3 slot for the CRU interface.

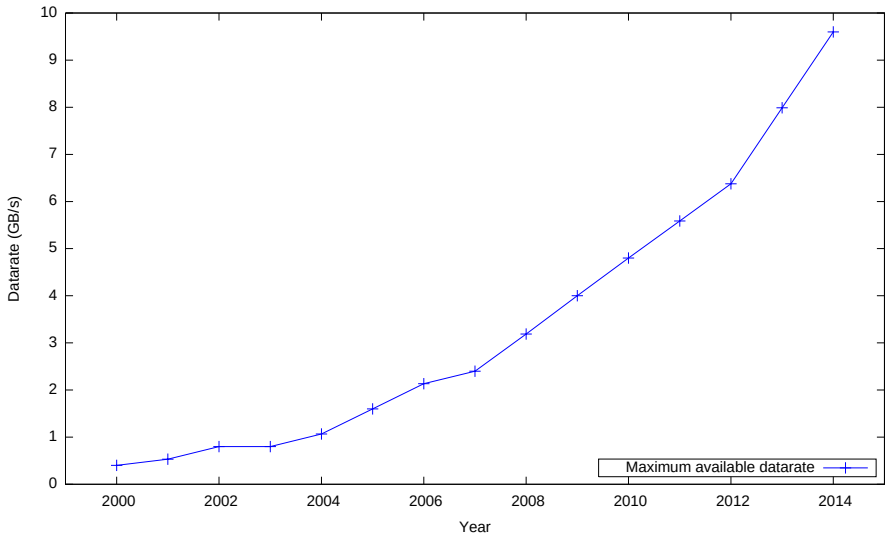

Figure 6.3: Evolution of maximum throughput to DDRx memory over the last years.

6.2.3 FPGA data flow online processing

FPGAs have a long history of data processing for High Energy Physics (HEP) experiments, from the handling of low-level protocols up to online processing and first event building tasks like clustering. Every new FPGA generation comes with an increased device size, and, as a consequence, a larger number of more complex algorithms can be implemented in the hardware to save processing resources later on. Examples include existing hardware cluster finders and algorithms for future data compression. Until now, these algorithms have been described using low-level hardware description languages like VHDL or Verilog. Low-level languages are well suited for describing interface blocks like PCIe, DRAM controllers or serial optical links. However, development is expensive in these languages for processing algorithms based on data flow or algorithmic level. Complex pipeline architectures with hundreds of stages can lead to code that is hard to read, while optimised processing modules with differing latencies cannot be integrated easily. Every new FPGA generation makes it harder to make efficient use of the available resources with low-level hardware description languages. Maintenance and modification of this kind of code is a complex task. Fortunately, in recent years, frameworks for pipeline generation have become available that can produce optimised pipeline architecture from a high-level data flow description and simplify development and maintenance of the code. Furthermore, several examples have shown that code generated from a high-level framework produces better results than handwritten code in a low-level language. Employing such techniques in HEP can dramatically reduce the design effort of developing firmware while producing more efficient hardware. First evaluations of a data flow implementation of the TPC fast cluster finder algorithm used during Run 1 have shown a comparable FPGA resource usage with a significantly reduced code volume compared to the plain VHDL implementation [3]. These techniques can be applied to preprocessing steps in an FPGA based RORC3 or as a co-processing component in the FLP nodes to ease the demands on FLP and EPN processing capacities.

6.2.4 CPUs

This section gives a short summary of the existing CPU architectures and discusses the perspectives for Runs 3 and 4.

– AMD Opteron / Intel Xeon: These x86 architectures have been the natural choice since they are widely supported and very generically usable.

  Xeon processors are typically used in 2 socket systems, with 4-18 cores per CPU today. Each core may run 2 threads. The number cores should increase to 28 cores by 2017 and 40 by 2019. New

CPU features are expected to include wider vector instructions going from 256 bits now to 512 bits later. Today's highest clock speeds are close to 4 GHz in the systems with fewer cores and are not likely to increase. On CPUs with more cores, the clock speed is actually reduced to $2-3$ GHz when all cores are active, dynamically boosting close to 4 GHz when thermal dissipation allows as when only a few threads are running. By 2018, the performance of a single core will therefore increase with new features like wider vectors rather than by higher clock speeds. The estimate of the performance of a mainstream CPU core in 4 years' time is based on the measurement of one today's top CPU core. Significant improvements in the performance of existing code will therefore be possible only by using the new features provided with each generation of CPUs.

– Atom: This is an x86-based architecture with a focus on low power consumption and low cost production. The architecture might therefore be an interesting alternative to the typical Intel Xeon / AMD Opteron CPUs.

  Although a single Atom core is much slower than a Xeon, the price/performance ratio can be up to 3 times better in some applications [4]. As Atom processors are smaller and optimised for power, they can be packed into a high density system, like the HP Moonshot (45 servers in 4.3 rack units).

– ARM: The main architecture used in mobile and embedded systems today. ARM systems have very low power to performance ratio, especially nodes with an integrated GPU.

– AMD Fusion: The AMD Fusion APU (Accelerated Processing Unit) is a high efficiency System on a Chip (SoC) combining an AMD x86 processor and an AMD GPU. The x86 CPU is derived from the Opteron designs but with fewer cores and smaller caches since the GPU part (see 6.2.5) is the primary processing device.

6.2.5 Accelerator architectures

This section gives a short summary of the existing GPU architectures and discusses the perspectives for Run 3.

– AMD Fusion: The GPU part of the Fusion SoC is derived from the standard AMD GPU cores. The difference is that both CPU and GPU have direct access to the host memory. Compared to traditional accelerators the memory is thus larger but slower.

– AMD GPU: This architecture provides thousands of low-power cores for highly parallelised performance and for offloading computation from the CPU. Programming is done via OpenCL or OpenACC compiler directives.

– NVIDIA GPU: This architecture provides thousands of low-power cores for highly parallelised performance and offloading computation from the CPU. Programming is achieved via CUDA, OpenCL or OpenACC compiler directives.

– Xeon Phi: This architecture is a low-power, many-core coprocessor for highly parallelised performance. The Xeon Phi can be used as an accelerator, similar to a GPU. In addition, it can be used as just another compute node in the system since it runs a Linux OS. For these reasons, it can be programmed both via offload models such as OpenMP or OpenCL, and directly in C++.

  The current Xeon Phi (Knights Corner, KNC) is a PCIe board with 54 cores. The next generation (Knights Landing, KNL) announced for end 2014 is planned to have 72 Atom cores (2-3x faster than existing KNC cores), able to run 4 threads per core, with 16 GB on-board MCDRAM working at 500 GB/s, support for DDR4, and 512-bits registers (AVX-512 instruction set). It will also be available as a standalone CPU (or coprocessor as now). According to [5], an integrated host interface fabric will be available for 2015, with on-board QSFP possibly up to 100 Gb/s.

6.2.6 Selection criteria

The selection of the hardware components will have to be made opportunely. Using a portable programming model can reduce the dependence of code development on specific hardware details. Deferring the hardware purchase will give the benefits of improved hardware functionality. The following hardware selection criteria will be used:

- Total Cost of Ownership: The combined cost of purchase, operation, and maintenance.
- Power Consumption of the system under typical workload.
- Programmability: Flexibility for selection of programming languages and frameworks which meet our software criteria.
- Reliability: Stability of device driver, stability of development tools, and expected failure rate of the hardware.
- Performance Benchmarks: The high-level performance benchmarks described in section 6.3 can be used to normalise the above criteria.

6.3 High-level benchmarks

The high-level benchmarks are based on typical reconstruction algorithms used by the ALICE experiment.

6.3.1 FPGA based Online Data Preprocessing

The FastClusterFinder algorithm, operating on TPC raw data, is an FPGA-based online processing core for extracting clusters and computing their properties [6]. The FastClusterFinder was successfully used in the HLT Read-Out Receiver Card (H-RORC) firmware as part of the HLT online reconstruction chain during Run 1. The FastClusterFinder is designed so that it can handle the full bandwidth of data transmitted via DDL. Its design follows the paradigm of a data flow architecture, streaming the data through the different processing sub-stages and performing all necessary calculations in parallel. Since there are no feed-back loops or branches, the design can be highly pipelined and parallelised. This allows a more efficient implementation than is possible on a CPU and easily compensates a CPU's significantly higher clock rates.

Figure 6.4 shows the performance of the FastClusterFinder compared to a software implementation. The measurements are based on about 5000 heavy ion event fragments recorded during Run 1. The blue points show the processing time required on a recent Ivy Bridge CPU running at 3 GHz for events of various input sizes. This is the same server hardware as used for the compute nodes in HLT during Run 2. The processing time is linearly dependent on the input data size. The hardware processing time is purposely limited only by the bandwidth of the input link. As the data are fed through the FPGA anyway, the FastClusterFinder only induces an additional latency of a few microseconds to the read-out. As DDL1 has only half the input bandwidth, its processing time is double that of DDL2.

The FastClusterFinder for DDL2 is approximately 25 times faster than the software implementation. To achieve the same processing rate with the software implementation, 25 CPU cores would be needed for each DDL link. Having 6 FastClusterFinder instances on each C-RORC for Run 2, each C-RORC saves 150 CPU cores for HLT.

There is a strong argument for using FPGAs in the data path because of their processing capabilities: they can significantly reduce the number of CPUs required in the O^2 facility. The baseline is therefore to use FPGAs for the TPC cluster finder. An FPGA with the adequate capacity (Altera Arria 10) is already available as an engineering sample.

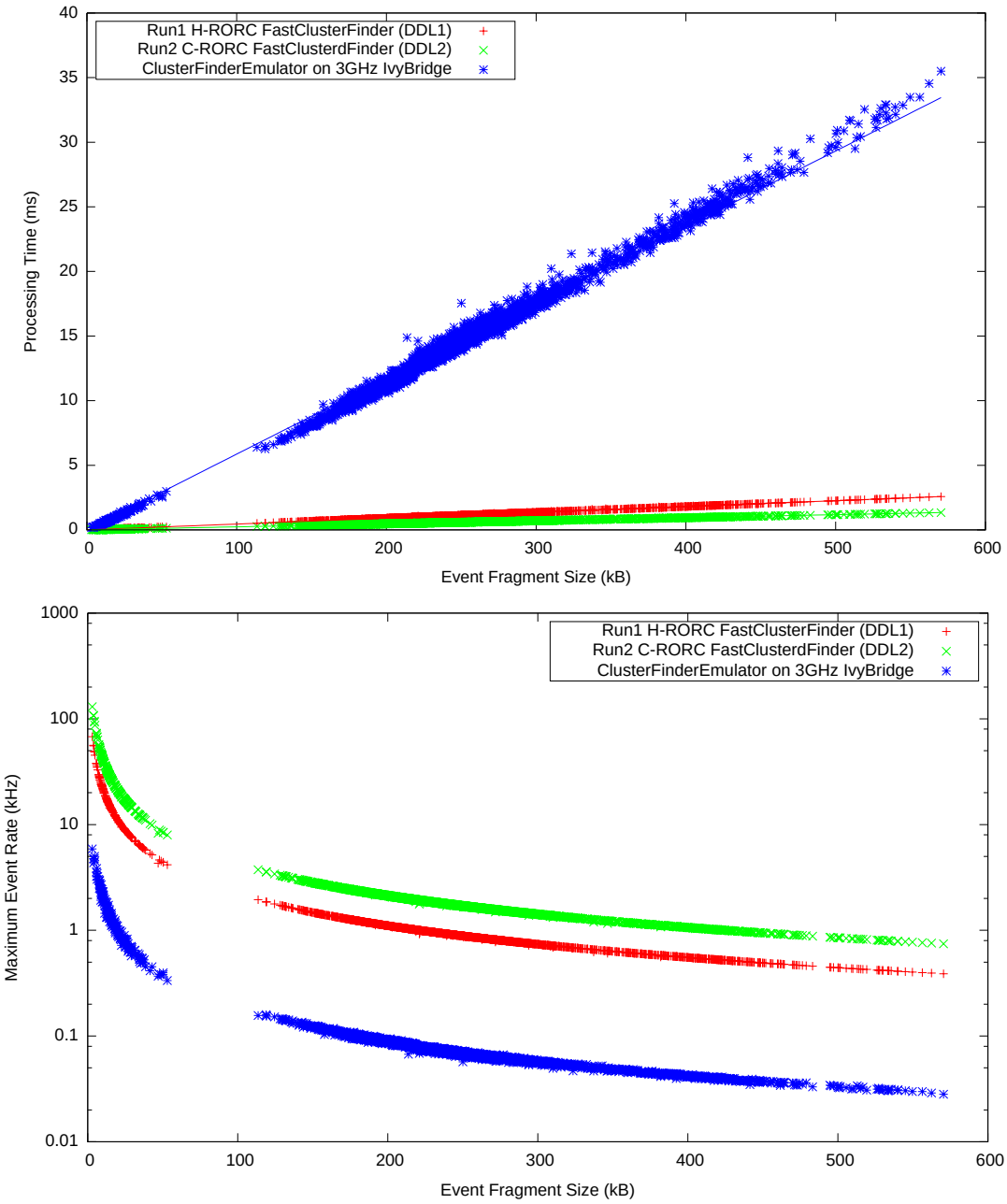

Figure 6.4: Performance of the FPGA-based FastClusterFinder algorithm for DDL1 as used during Run 1 (red +) and DDL2 running at 4.25 Gb/s (green x) compared to the software implementation on a recent server PC (blue *). The plot on the top of the figure shows the required processing time relative to event size. The one on the bottom of the figure shows the same measurements as rate equivalent.

6.3.2 HLT TPC Track Finder

The HLT TPC Track Finder uses the Cellular Automaton principle to construct track seeds and then uses (a simplified) Kalman filter to fit tracks to the seeds and extend the seeds to full tracks in the TPC volume. It is implemented for OpenMP (CPU), CUDA (Nvidia GPU), and OpenCL (AMD GPU), with all versions sharing a common source code. The Track Finder process consists of many sub-steps, each with its own computational hotspots, making it a benchmark covering a mixture of compute throughput, memory bandwidth, and memory latency. The benchmark permits to compare the execution time on different GPU and CPU systems, as shown on Fig. 6.5.

Table 6.3 shows the performance measured on several CPU and GPU platforms.

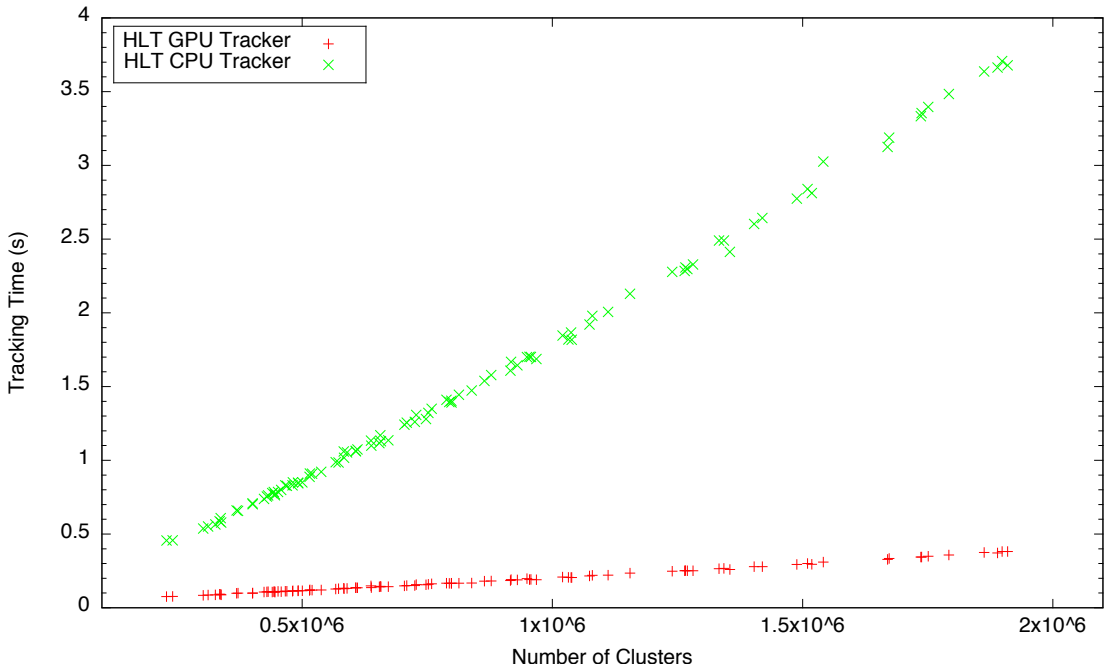

Figure 6.5: Tracking time of HLT TPC CA tracker on Nehalem CPU (6 Cores) and NVIDIA Fermi GPU.

6.3.3 HLT TPC Track Fitter (HLT Track Merger)

The HLT TPC Track Fitter (HLT Track Merger) runs on top of the HLT TPC Track Finder and performs a refit of the tracks by use of the Kalman filter. It is implemented in OpenMP (CPU) and CUDA (Nvidia GPU). An OpenCL version will be developed for performance analysis with AMD GPUs. In contrast to the Track Finder, it uses the full Kalman filter making it mostly limited to compute throughput.

Table 6.4 shows the performance measured on several CPU and GPU platforms.

The baseline is to use GPUs for the TPC tracking. Some existing GPU cards (AMD S9000) already have the adequate performance for this application.

6.3.4 ITS Cluster Finder

The ITS Cluster Finder algorithm for the Inner Tracking System detector identifies and computes the centre of gravity coordinates of adjacent hits on the pixel chips, grouped into clusters. Input data consists of a list of hits a few hundred per chip: integer 2D coordinates, ordered by row and column. Output data are a list of clusters of a few tens per chip: floating point 2D coordinates. This algorithm is interesting because of the small input/output data sets ($\sim$1 kB), the ease of implementation and its independence from any external libraries. It can therefore be ported to a number of different devices, including the ones with limited resources. The algorithm can easily be run in parallel with one pixel chip processed per thread. Vectorisation is quite limited due to the serial nature of the operations to be performed on incoming data; decisions on a pixel depend on what comes next in the stream. Implementations for x86 and GPU exist and can be a good basis of comparison for current and future devices with this type of workload. Fig. 6.6 shows the performance on a dual socket IvyBridge server, capable of 7 kHz on the simulated data sample from 430 inner chip modules. This represents around 1/50th of the full detector but a larger fraction of the data, as the occupancy greatly decreases in the outer modules.

The same algorithm running on the full detector data set, assuming noise level of 10^{-5}, is able to process events at 500 Hz on the 12-core reference server. Around 1200 cores would therefore be needed for the

ITS cluster finder for Pb-Pb at 50 kHz.

Compared to the x86 ITS cluster finder, the GPU implementation is one order of magnitude slower for this computing task which has too few floating point operations for this architecture. On the other hand, the algorithm benefits from the random execution and pipelining on the x86.

The baseline is to use the FLP CPUs for the ITS cluster finder. The computing performance of the ITS FLPs is tailored to this task with more cores than the other FLPs.

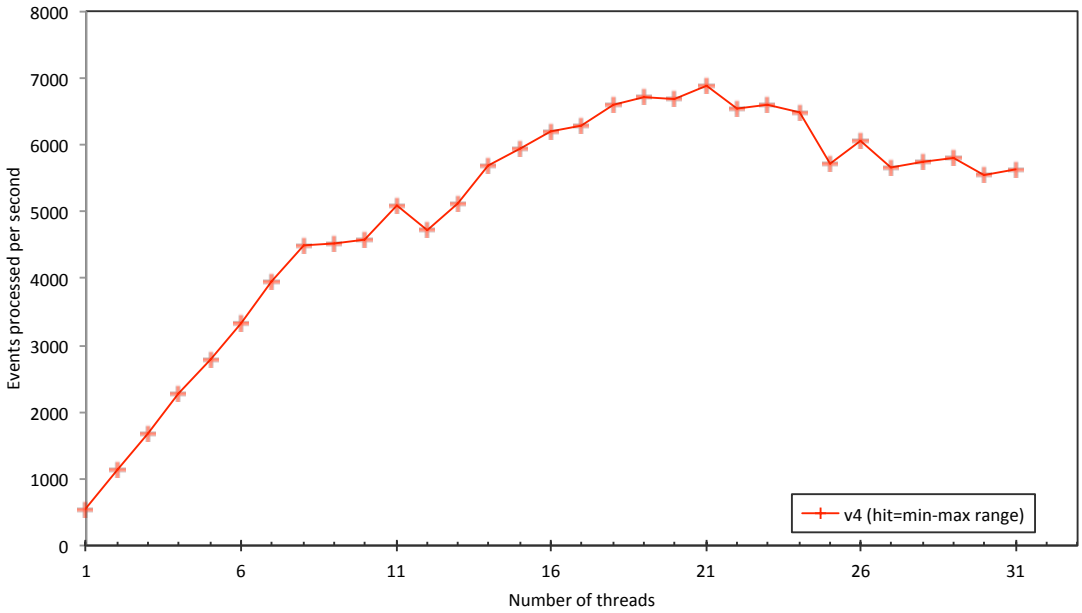

Figure 6.6: ITS cluster finding performance on simulated data for 430 inner chip modules, as a function of the number of processing threads running.

6.4 Low-level benchmarks

The low-level benchmarks are not intended to select the fastest hardware but to identify possible bottlenecks of the hardware components in a controlled environment.

6.4.1 PCIe benchmark

The PCIe benchmark (see Tab. 6.2) shows that PCIe yields a single-duplex bandwidth of 75-90% of the specified peak bandwidth which is sufficient for the ALICE workload. This benchmark is not intended to find the hardware with the highest peak bandwidth, but to exclude hardware that cannot deliver sufficient bandwidth under the following conditions:

- In full duplex mode
- Using multiple PCIe cards (at the same and at different root-complexes) to achieve full performance in concurrent transfers
- With non-linear transfers (i.e. submatrix transfers)
- In a system with high memory load caused by other applications

Table 6.2: Result of the PCI Express Benchmark dual 8-core Sandy-Bridge System. Implemented in CUDA / OpenCL.

	Unidirectional Bandwidth (1 GPU) (GB/s)	Aggregate Bandwidth (GB/s)
4 × GTX580 (Half Duplex, PCIe Gen2)	3.63	29.1
2 × S9000 (Full Duplex, PCIe Gen3)	7.90	31.6

6.4.2 Compute benchmarks (DGEMM / Matrix lib based)

Optimised DGEMM (Matrix-Matrix Multiplication) implementations usually achieve close to peak performance making this a simple method to test the peak achievable compute throughput under optimal conditions. Most vendors deliver optimised DGEMM libraries, so only a small effort is required to run this benchmark. Table 6.5 shows the performance measured on several CPU and GPU platforms.

6.4.3 memcpy() benchmark

The memcpy() benchmark is a simple yet valid test to measure the speed reached when copying a large ($>$ 1GB) chunk of memory. This metric has proved to be useful for estimating the global throughput in intra- or inter-thread communication. It helps to plan the balance between I/O and CPU resources. On i5-680 at 3.7 GHz and X5677 at 3.7 GHz, a throughput of 2.1 GB/s was measured. Experience shows that at least 3 threads are necessary to saturate a 40 Gb/s link.

6.4.4 NUMA benchmark

This benchmark was developed to execute different memory usage patterns on all possible memory addresses in a system [7]. For a NUMA system, it will show the different NUMA regions in the address ranges and how much the CPU interconnect lowers the available memory performance. Figure 6.7 shows the total load & store bandwidth for a test that reads values from memory, adds 1 to the values, and stores them back to the same address. Every value has to move over the memory bus twice: once for the load and a second time for the store. A bandwidth of 40 GB/s thus implies that 20 GB of memory were modified in one second.

6.5 Performance conversion factors

Conversion factors are used to estimate the performance gain of multi-core (with Hyperthreading) or GPU implementations. In each case two factors are defined representing the performance and the CPU usage increase. The factors are defined as follows:

- Speedup of Multithreading
  With n_1 the runtime on 1 core and n_2 the runtime on x cores, the factors are defined as $\frac{n_1}{n_2}$ and $\frac{x}{1}$.

- Speedup of Multithreading with Hyperthreading
  With n_2' the runtime with x' threads on x cores ($x' > 2x$), the factors are defined as $\frac{n_1}{n_2'}$ and $\frac{x}{1}$. We consider that this will not scale linearly, but it is needed to compare single-thread performance to full CPU performance.

- Speedup of GPU (Factor vs x', Full CPU)
  The GPU will be compared to the full CPU, and the comparison shall also take into account the CPU resources occupied by the GPU implementation. With the GPU implementation requiring y CPU cores and having a runtime of n_3, the factors are $\frac{n_2'}{n_3}$ and $\frac{y}{x}$.

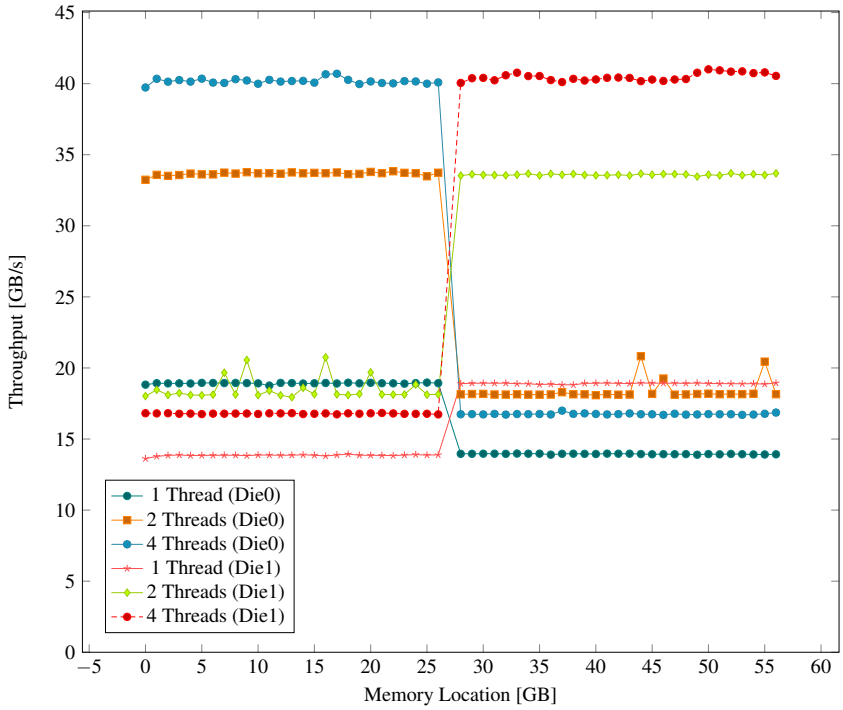

Figure 6.7: Throughput of the "Add One" test of NUMAbench on a dual socket Intel Xeon SandyBridge system with 64 GB RAM (DDR3 1600 MHz).

– Speedup of GPU (Factor vs 1, how many CPU cores does the GPU save)
  Using y CPU cores the GPU achieves a speedup of $\frac{n_1}{n_3}$ thus saving $\frac{n_1}{n_3} - y$ cores. The factors are $\frac{n_1}{n_3}$, $\frac{y}{1}$ and $\frac{n_1}{n_3} - y$.

These factors are shown for in Tab. 6.3 for the TPC Track Finder benchmark, in Tab. 6.4 TPC Track Fit Benchmark and in Tab. 6.5 for the Matrix Multiplication Benchmark.

6.6 Programming models

6.6.1 Compute-Kernels

Since computational hotspots are usually limited to small parts of the code, it is possible to use different paradigms for code of different computational effort. Given this heterogeneous platform, it is reasonable to keep generic code, running on standard processors. Computational hotspots, however, should use kernel code[1] for parallel hardware to achieve a general application speedup. The programming and hardware models for this kernel code must be investigated thoroughly.

Generic code, requiring little computational effort, must focus on maximum readability and maintainability, while code with larger computational effort must be written considering vectorisation and parallelisation (Fig. 6.8). Nevertheless, all code should use data structures that have been defined with parallelisation in mind in order to avoid additional conversions. Kernel code, being burdened with extreme computational effort, must follow the strictest guidelines to make it compatible with the restrictions imposed by the languages and libraries providing the most computational power (Fig. 6.9).

[1] "kernel code" refers to a computational kernel, not the OS kernel

Table 6.3: Results of the TPC Track Finder benchmark implemented on different computing platforms from different vendors: in OpenMP for CPUs (upper part) and CUDA / OpenCL for GPU (lower part).

CPU platform	Clock (GHz)	Number of cores	Number of threads	Hyper-threading	Time per Event (ms)	Factors
Intel Nehalem (Smaller Event)	3.6	4	1		3921	
			4		1039	3.77 / 4
			12	$x = 4, x' = 12$	816	4.80 / 4
Intel Westmere	3.6	6	1		4735	
			6		853	5.55 / 6
			12	$x = 4, x' = 12$	506	9.36 / 6
Intel Sandy-Bridge (Dual socket)	2.0	2×8	1		4526	
			16		403	11.1 / 16
			36	$x = 16, x' = 36$	320	14.1 / 16
AMD Magny-Cours	2.1	2×12	36	$x = 24, x' = 36)$	495	

GPU platform	CPU cores used by the GPU for data transfer	Time per Event (ms)	Compared to Sandy-Bridge System Factor vs. x′ (Full CPU)	Factor vs. 1 (1 CPU Core)
NVidia GTX580	3	174	1.8 / 0.19	26 / 3 / 23
NVidia GTX780	3	151	2.11 / 0.19	30 / 3 / 27
NVidia Titan	3	143	2.38 / 0.19	32 / 3 / 29
AMD S9000	3	160	2 / 0.19	28 / 3 / 25
AMD S10000 (Dual GPU)	6	85	3.79 / 0.38	54 / 6 / 48

Table 6.4: Results of the TPC Track Fit Benchmark on different computing platforms from different vendors: in OpenMP for CPUs (upper part) and CUDA / OpenCL for GPU (lower part).

CPU platform	Clock (GHz)	Number of cores	Number of threads	Hyper-threading	Time per Event (ms)	Factors
Intel Westmere	3.6	6	1		125	
			12	$x = 6, x' = 12$	17	7.36 / 6

GPU platform	CPU cores used by the GPU for data transfer	Time per Event (ms)	Compared to Sandy-Bridge System Factor vs. x′ (Full CPU)	Factor vs. 1 (1 CPU Core)
NVidia GTX580	0	7	2.5	18.4 / - / 18.4

Table 6.5: Result of the Matrix Multiplication Benchmark on different computing platforms from different vendors: in OpenMP for CPUs (upper part) and CUDA / OpenCL for GPU (lower part). DGEMM scales completely linearly on CPUs, only GPU factors are stated.

CPU platform	Clock (GHz)	Number of cores	Number of threads	Hyper-threading	Time per Event (GFLOP/s)	Factors
Intel Sandy-Bridge	2.2	8			180	
AMD Magny-Cours	2.1	12			270	

GPU platform	CPU cores used by the GPU for data transfer	Time per Event (GFLOP/s)	Compared to Sandy-Bridge System Factor vs. x′ (Full CPU)	Factor vs. 1 (1 CPU Core)
AMD S10000	0	2900	10.7 / 0.5	

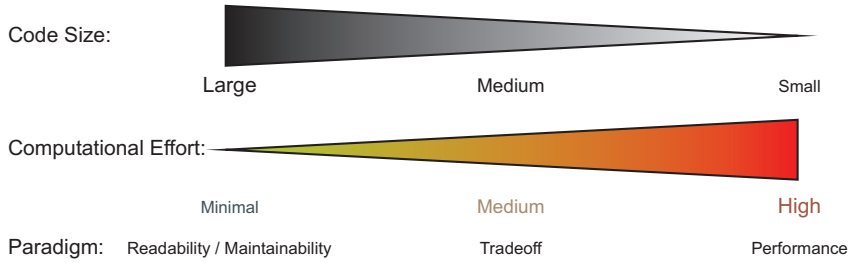

Figure 6.8: The computational hotspot often lies within a few lines of code, while the majority of the code is largely irrelevant from a performance perspective.

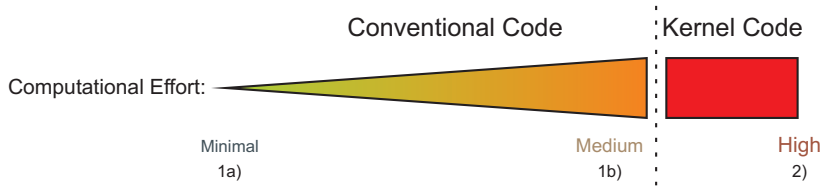

Figure 6.9: Separating the hotspots into kernel code allows to differentiate paradigms.

6.6.2 Programming languages and frameworks

Some languages and frameworks are presented below. The focus is on known high performance computing applications suitable for the ALICE upgrade. Although not exhaustive, this selection would permit the development of software for all the types of hardware described in section 6.2.

Languages & frameworks for compute-kernels

The following languages and frameworks are being considered to be used in the O^2 project:

- CUDA / OpenCL : These are considered the two most important languages for GPU (accelerator) programming with low-level control. Since it is possible to write kernel code that is straightforward to convert between CUDA and OpenCL, the two are listed as one item. CUDA is a set of language extensions and directives for management of NVIDIA GPUs [8]. OpenCL is a portable, open, royalty-free language standard plus Application Programming Interface (API) for parallel programming of heterogeneous systems including GPUs and other devices [9].

- OpenACC / OpenMP 4 : It is anticipated that OpenACC [10] will be incorporated within future releases of OpenMP [11]. OpenACC is an open, directive-based standard for parallel programming of CPU and GPU systems, with future support of Xeon Phi anticipated. The OpenACC programming model is relevant because small problems can be brought to the accelerator via directives placed within the high-level language (such as C++). OpenMP is a set of language extensions plus API for multi-threading, or shared memory multiprocessing, on multi-core CPUs and accelerators.

- Vc : The Vc library enables expression of data-parallelism used for vectorisation in a portable and efficient way. It does so by extending the C++ type system with the necessary types to abstract SIMD hardware features. The library is important for high performance code on the CPU and makes porting to the Xeon Phi easy.

– Upcoming C++ concurrency/parallelism technical specifications (TS) : The C++ standards committee is investing in native support for (highly) parallel code that can (in theory) be executed on accelerators. This can supersede developments like C++AMP in a portable way. A TS for this work may appear in 2015.

The following languages and frameworks have been excluded:

– OpenGL compute : This interface is most interesting for code that is already based on OpenGL. No further investigation was done to test its applicability in ALICE workloads.

– OpenHMPP : OpenHMPP only has proprietary compiler support. OpenACC is the better alternative.

– OpenMP3 : OpenMP in version 3 does not have support for offloading to accelerator hardware or SIMD loops. OpenMP 4 will be supported in upcoming C++ compilers and is the preferred version.

– C++AMP : This interface is available for Windows only. Alternatives are in development for Standard-C++ or via directives such as OpenACC and OpenMP 4.

Languages & frameworks for generic code

– C++11 : Most of the generic code will be written in standard C++. We require the C++ version to be at least C++11 because it introduced standardized thread support and a suitable memory model.

– Vc : Vc will be important to create data-structures suitable for vectorisation on the CPU. Also, some of the generic code may be vectorised, and Vc provides an easy interface.

– OpenMP4 : OpenMP can be used as a portable solution for parallelised generic code. Especially if OpenMP 4 is used for kernel code, it will be a natural choice throughout the whole code base.

– OpenCL : OpenCL is primarily considered for kernel code; however, since it can also target CPUs, it might be useful for generic code, too.

6.7 Networking

Today's high performance networking technology is dominated by two standards: Ethernet and Infiniband (IB). Ethernet Network interface cards are available for 10 and 40 Gb/s. Transferring data over an Ethernet network using the TCP/IP protocol requires a substantial CPU capacity: a data transfer at a sustained bandwidth of 10 Gb/s uses one CPU core at 50% and the full use of a 40 Gb/s port requires 2 cores. IB Host Channel Adapter (HCA) are available for 10, 20, 40 and 56 Gb/s. Data transfer over an IB link using the TCP/IP protocol (IP over IB or IPoIB) is also using a substantial amount of CPU. The Remote Direct Memory Management (RDMA) has been developed to reduce the CPU usage. RDMA allows server to server data movement directly between application memory without any CPU involvement reaching 80% of the nominal IB Fourteen Data Rate (FDR) performance as shown in Tab. 6.6.

Table 6.6: Result of the Infiniband and Ethernet physical layer network benchmarks. Tested with Mellanox ConnectX3 on dual-socket Sandy-Bridge.

	Maximum bandwidth (GB/s)	Measured bandwidth (GB/s)
QDR - Native IB Verbs	4.0	3.9
FDR - Native IB Verbs	6.8	5.6
FDR - IPoIB TCP Transfer	6.8	2.5
Eth 40 Gb - TCP	5.0	4.9

Both standards have plans to increase their maximum bandwidth to 100 Gb/s and above [12] [13]. A new technology has been recently announced by Intel, the Omni-path, aiming at bandwidth of 100 Gb/s and more [14]. This technology initially available as discreet components in 2015, will also be integrated into the next-generation Intel Xeon and Xeon Phi processors providing more effective direct I/O into the processor.

There are already two technologies with the adequate performance for the O^2 system and there will probably be a third one by the time of purchase. Given the foreseen evolution, no decision is made at this point and the software framework will be developed in a way independent from any network technology.

6.8 Data storage hardware

The O^2 local data storage system must be scaled to absorb the maximum data rate which is achieved at the beginning of an LHC fill. Two points must be taken into consideration: the storage type with its performance and its attachment to the data initiator. For attachment types that have enough storage bandwidth to absorb the rate, the following technologies are considered acceptable:

- SAS: 12 Gb/s
- FiberChannel: 16 GB/s, 32 GB/s in preparation
- ISCSI: depending on network speed: 10 Gb/s (Ethernet), minus protocol overhead
- IB: 40 Gb/s for DAS, minus protocol overhead

Today's 10 kRPM SAS disks can exceed a sustained throughput of 120 MB/s, so with relatively few disks in a logical volume (RAID set, Dynamic Disk Pools (DDP)) substantial performance can be achieved; 8 disks in a RAID-6 set can already deliver more than 1 GB/s as shown on Fig. 6.10. The performance of a logical volume can be increased by adding disks to a maximum of about 20. With more disks, the setup becomes very inefficient.

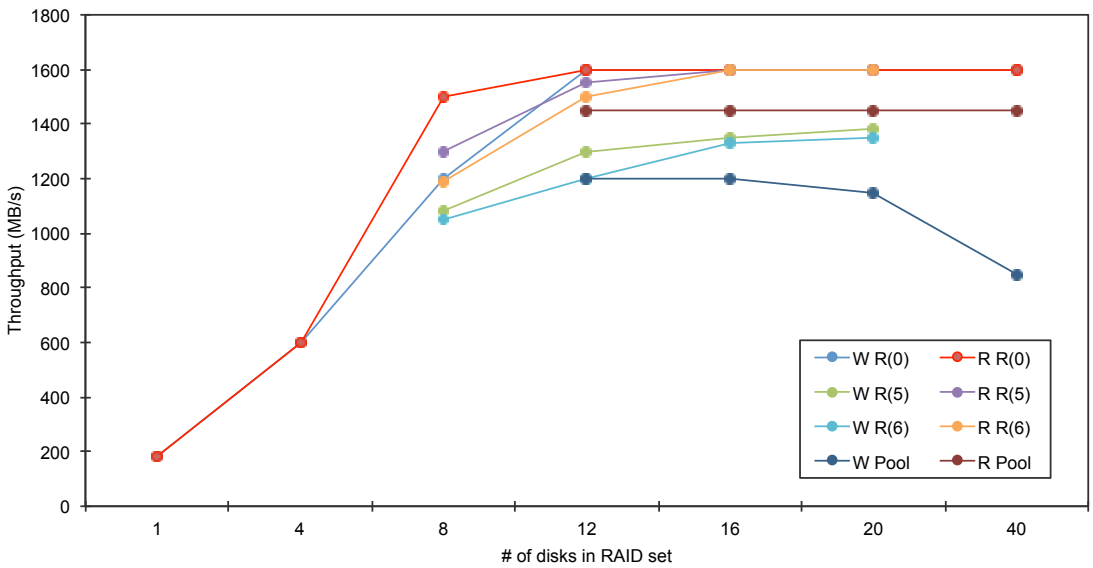

Figure 6.10: Performance using different number of disks in RAID set.

The block size used for storage transactions is another critical factor. See Fig. 6.11 showing the storage throughput versus the block size for three different common file systems on the same storage volume.

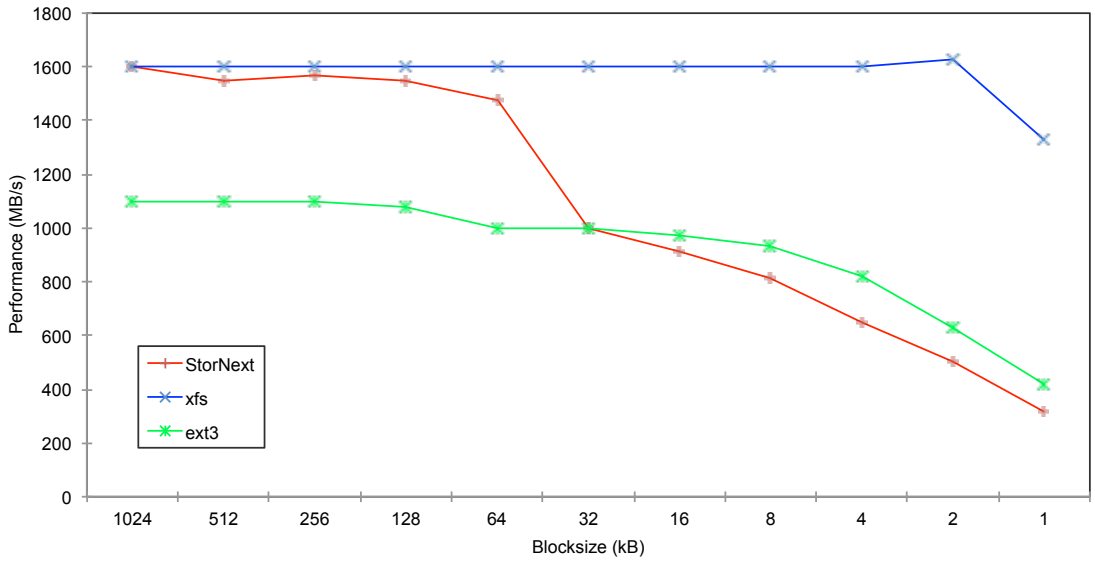

Figure 6.11: Storage performance using different block sizes.

These measurements were taken on a pure flash storage system optimised for full throughput down to 2 kB blocks. The present ALICE DAQ storage system has already more than 20 GB/s data throughput. Future developments will almost certainly provide easier and more compact solutions.

Another critical requirement is that the O^2 data storage capacity has to be large enough to store the data of a complete year of ALICE data. At the time of writing, the first SATA archive disks with a storage density of 1 Tb/in^2 for a total capacity 8 TB are on the market. The enterprise quality SAS disks used for demanding applications have a total capacity of 6 TB. The trend of research is clearly to find ways to increase storage density to 4 Tb/in^2 by 2021 [15] [16]. If industry predictions (see Fig. 6.12) are correct, hard disks with an increased storage capacity of a factor 1.5 to 3 will be available at the time of procurement.

6.9 Technology bets

This section summarizes in Tab. 6.7 the bets which are done for the various technologies used by the O^2 system.

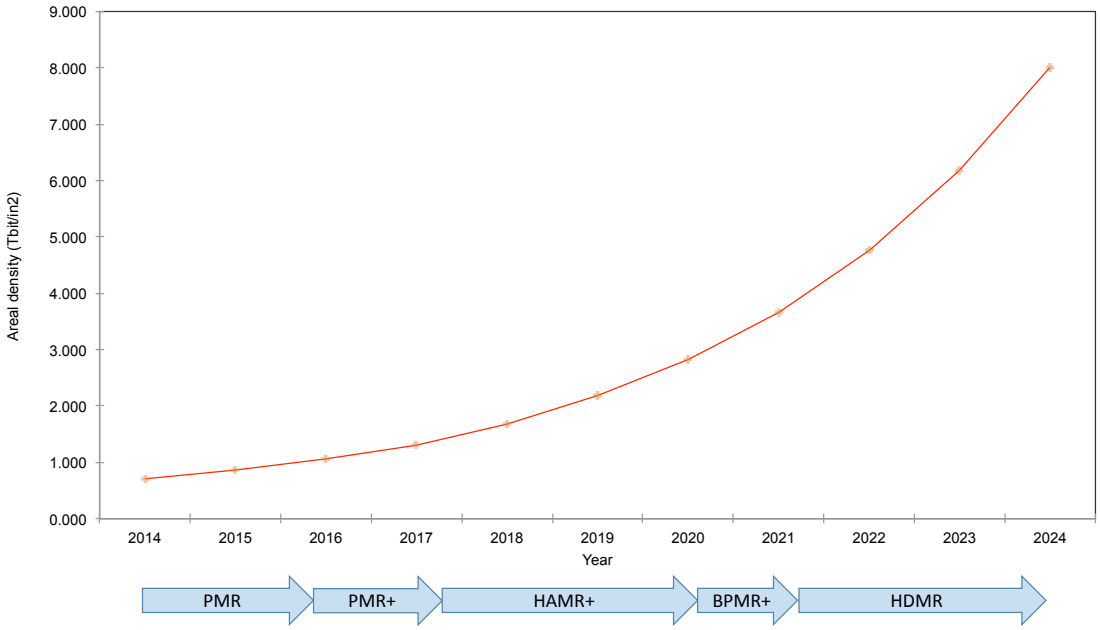

Figure 6.12: Prediction of hard disks storage density by the Advanced Storage Technology Consortium (ASTC). The evolution as a function of time shows the impact of the successive technologies: Perpendicular Magnetic Recording (PMR), PMR with Two Dimensional Magnetic Recording (TDMR) and/or Shingled Magnetic Recording (SMR) (PMR+), Heat Assisted Magnetic Recording with TDMR and/or SMR (HAMR+), Bit Patterned Magnetic Recording (BPMR) with TDMR and/or SMR (BPMR+) and Heated-Dot Magnetic Recording (HDMR).

Table 6.7: Technology future bets.

Technology	O^2 need	Availability	Risk
Input-output	77 Gb/s I/O slot	PCIe Gen3 x16 is available now	None
FPGA	24 GBT receivers and 24 cluster finders	Arria 10 Gx engineering sample	Very low: the chip expected to be commercially available in the coming months
CPU	EPN CPU with 32 cores	Currently maximum 18 cores	Low: CPU chips expected to include 28 cores by 2017 and 40 by 2019
GPU	TPC track seeding and following in less than 0.1 s	AMD S9000 is available now	None
Network	Port bandwidth of at least 40 Gb/s	Available now in 2 technologies and probably 3 at the time of purchase	None
Data storage	Bandwidth write and read: 90 GB/s	Possible with existing equipment	None
	Storage capacity: 70 PB	Current hard disks of 8 TB	None on the technology but a bet on the availability of 20 TB hard disks

6.10 Data storage software

6.10.1 Cluster file system

Several solutions are envisaged for accessing data storage hardware by all EPNs and DMs. Network Attached Storage (NAS) systems could provide the required functionality but impose severe restrictions on performance and redundancy, for example the Network File System (NFS). For this reason ALICE is considering different approaches such as the Cluster File System (CFS), already used during Run 1. The architecture of a CFS differs from the traditional NAS by the fact that all clients can access all storage elements directly without directing the data traffic through a central server.

For CFS the central server plays the role of a Meta Data controller (block allocation, file directory) while the clients access the storage elements through the network. The CFS architecture is a long standing industry standard for all data throughout intensive applications. For Runs 1 and 2, ALICE chose Quantum StorNext as the CFS base as it is a mature product. It meets present performance needs and could also be used for Runs 3,4 and , but the cost and structure of its underlying Fiber Channel network is no longer feasible with the number of file system clients needed in the future. With the arrival of Big Data and the evolution of networks and storage technologies over recent years, other interesting products became available that are also suited to address the requirements of ALICE for Runs 3,4 and . In particular the following two solutions are good candidates for ALICE.

Lustre file system

Originating from an open source development from 2001, the development of this network CFS is now based with INTEL which established it as a file system for the HPC and Big Data worlds. One advantage of Lustre is the support of different network and storage technologies. The minimal building block consists of two Object Storage Servers (OSS) providing redundancy for storage access connected to a common network with the clients as shown in Fig. 6.13.

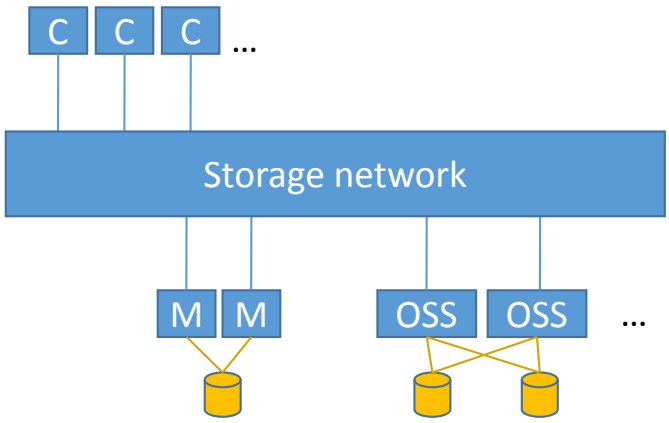

Figure 6.13: Building blocks of a Lustre system including Clients (C), MetaData Servers (M) and one OSS stack made of two cross-linked OSS servers.

Performance tests of a minimum Lustre configuration as shown in Fig. 6.13 have been performed. The setup consisted of two OSS servers connected to a IB FDRnetwork and both having dual 12 Gb/s SAS connections to disk arrays of 120 hard disks. Figure 6.14 shows good block size performance for ALICE data block sizes which are larger than 256 kB.

Figure 6.15 shows that the write performance of one OSS stack exceeds 4 GB/s, limited by the disk controller.

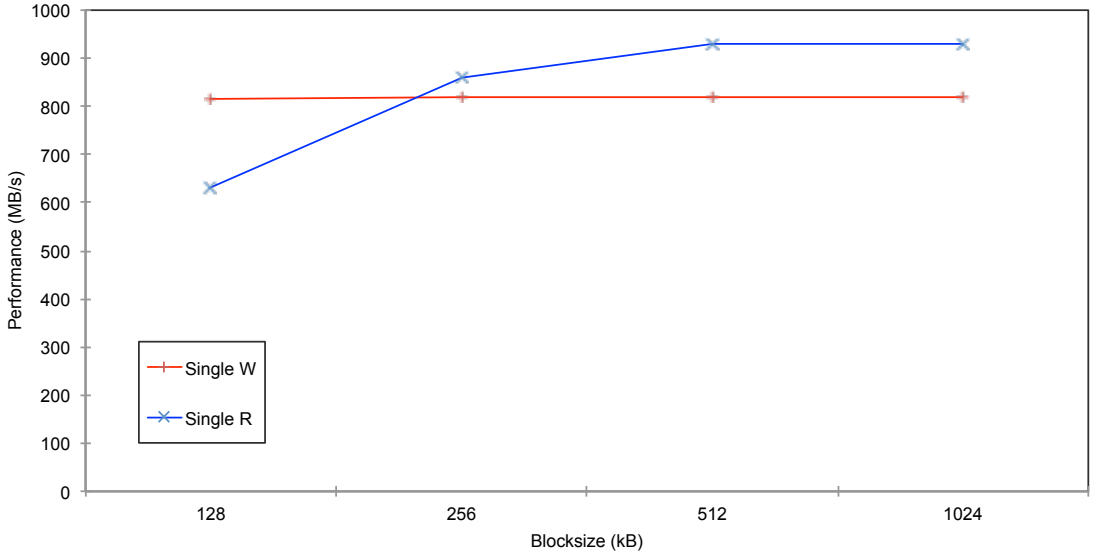

Figure 6.14: Lustre performance for one client using one stream with different block sizes (Write: +, Read: x).

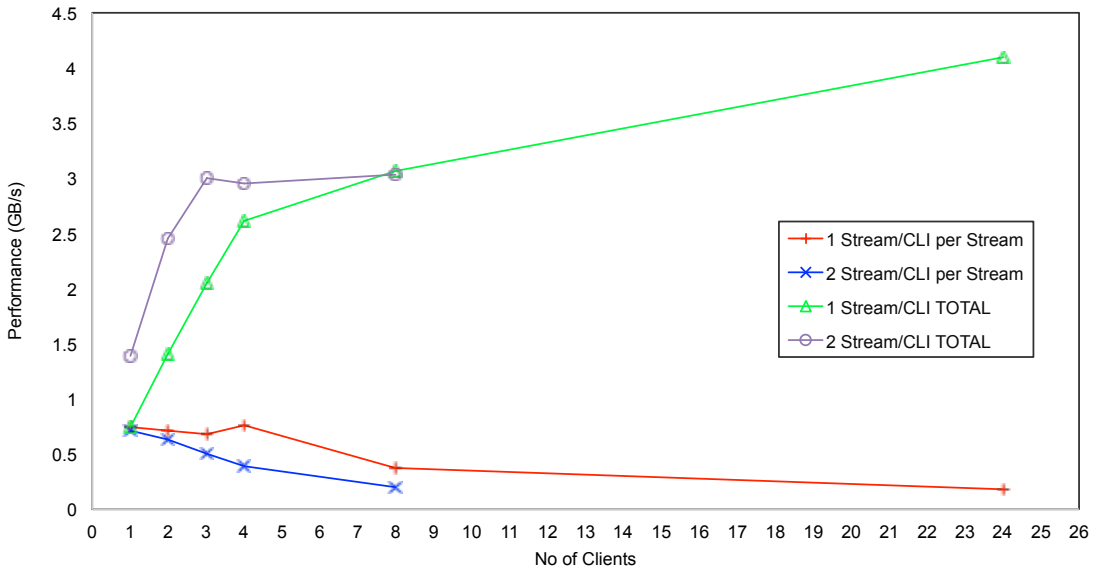

Figure 6.15: Lustre performance for multiple clients using 1 or 2 streams (Write performance per stream : +, Total write performance: △, Write performance per stream for 2 streams: x, Total write performance for 2 streams: ○).

Figure 6.16 shows that the Read performance of one OSS stack exceeds 11 GB/s (limited by the OSS network adapter) and that the performance increases with the number of clients using the same OSS. As OSS stacks work independently and have direct connections to the client, the total system performance can then be increased by adding further OSS stacks.

Figure 6.17 shows that on a global filespace the write and read performance per client exceeds 2 GB/s while maintaining more than 250 MB/s per stream.

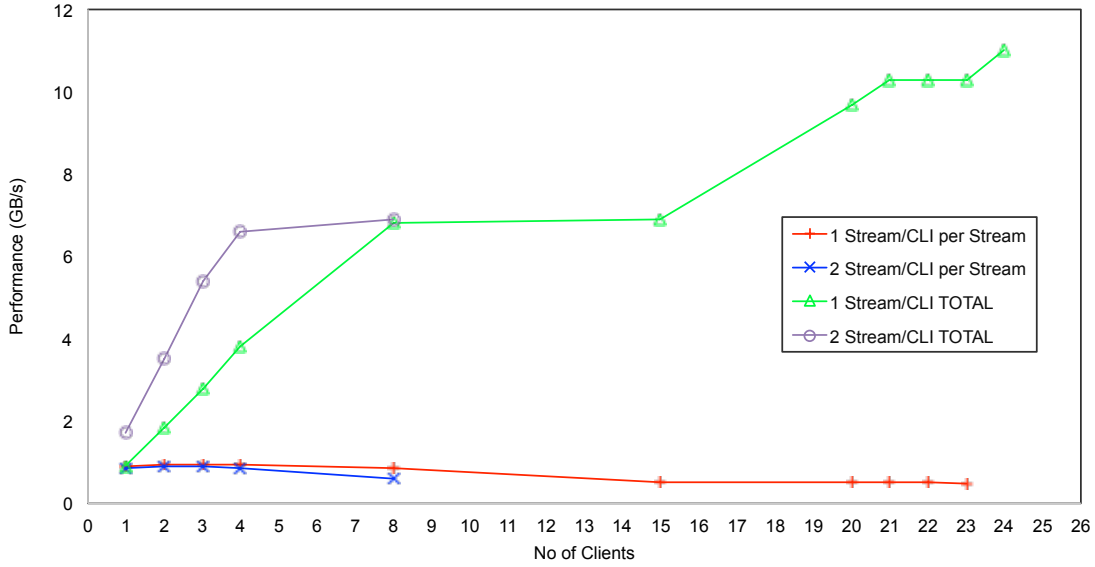

Figure 6.16: Lustre performance for multiple clients using 1 or 2 streams (Read performance per stream : +, Total Read performance: △, Read performance per stream for 2 streams: x, Total Read performance for 2 streams: ○).

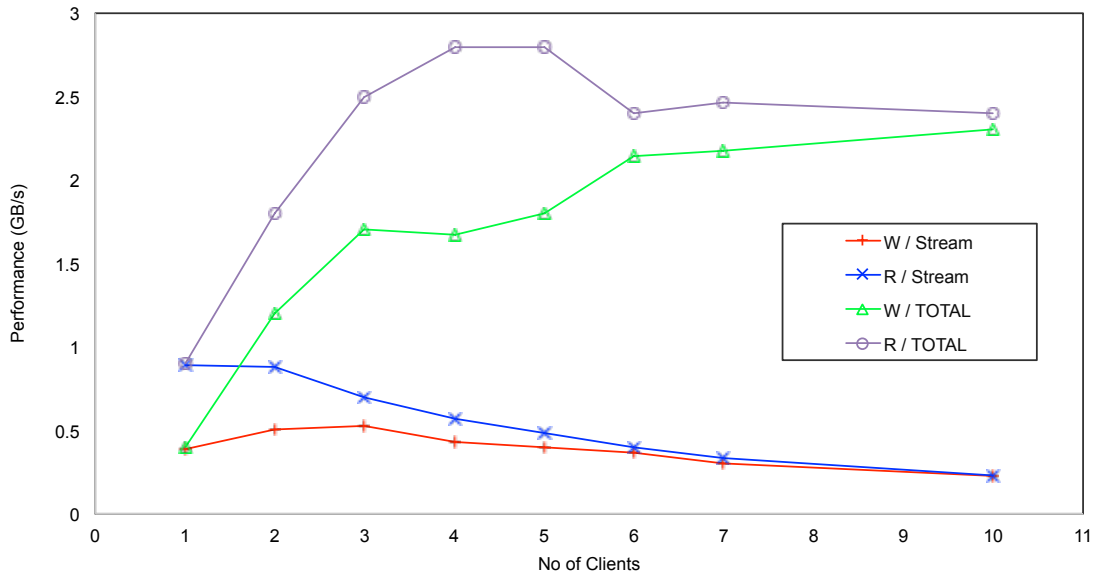

Figure 6.17: Lustre performance for 1 client using multiple streams (Write performance per stream : +, Read performance per stream: x, Total write performance : △, Total read performance : ○).

IBM GPFS

Available since 1998, this massive scalable file system is used by several TOP500 installations and has also been chosen by DESY for their data storage. The overall architecture is comparable with LustreFS with the exception of the absence of meta data servers as the block distribution is managed by the storage nodes (thus eliminating a point of failure and allowing for absolute linear scaling with the number of storage stacks). Performance of one storage stack exceeds 10 GB/s for read and write. GPFS provides

several unique features, particularly interesting are the usage of a DDP like data structure to boost write performance and to reduce rebuild times in case of drive failure and a full end-to-end protection against silent data corruption (bit flips, off-track, lost writes). This file system is definitively another candidate for ALICE and in-depth tests are foreseen for 2015.

6.10.2 Object stores

"Object stores" are storage back-end that address high storage demands for unstructured and immutable data. These stores have already reached a capacity of tens of petabytes. They are installed as production systems by web and cloud companies for the storage of videos, pictures, backups, or saved game states [17–20]. Apart from these customised object stores, commercial providers sell object store software at the petascale [21–24].

Object stores have a flat namespace and a simple interface, usually HTTP-based, that supports the addition of pairs of unique keys and retrievable large binary objects (BLOBs). In this way, they differ from the hierarchical data organisation found in distributed file systems as well as from a data model of tables and mutable rows found in NoSQL databases. The simple data model of object stores permits easy scaling of the aggregated volume and throughput.

Object store system blueprints and lessons learned from operational experience were published for Microsoft Azure Storage [19], a 70 PB, and for Facebook f4 [18], a 65 PB storage system. Microsoft declared a sustained disk bandwidth utilisation of 70% of the raw capacity in 2011. Both systems use Reed-Solomon erasure coding [25], thus keeping the replication factor at $\sim$1.5 with the same or better fault tolerance guarantees than three independent data copies. The data placement and failure response is centrally managed by a coordinator, which is a highly available cluster of a few nodes either in a primary backup configuration or by a distributed consensus protocol [26–28]. Similarly, the index of stored keys and their associated metadata is kept in a central, easily accessed NoSQL database. Additional compute-heavy nodes are deployed for regular data health checks and re-computation of redundancy blocks upon hardware faults.

An open-source object store with support for erasure coding is provided by RADOS [29], an independent base layer of the Ceph file system [30]. Several such deployments are at data centres within the LHC computing grid (e. g. CERN, RAL, BNL).

A major benefit of using an object store for the O^2 system is that the input comes in large, equally sized 6 GB files that will not need to be repackaged into larger units. A modest amount of metadata is produced which has to be managed. A storage system providing 75 PB of logical storage can store at most 12.5 million 6 GB files. So the index of files plus a few hundred bytes of metadata per file fit into the DRAM of a single server. The rate of writing new metadata is well within the limits of highly available consistent tools such as ZooKeeper [27].

With 10+4 Reed-Solomon erasure coding, a minimum of 840 hard drives at 100 MB/s is required to sustain the input rate of 60 GB/s. Before writing to hard drives, input data needs to be striped and redundancy stripes need to be computed. A standard multi-core CPU can encode more than 1 GB/s of input data [31, 32]. Assuming a packaging of 12 hard drives and two 10 Gb network adapters per node, 60 GB/s of input files can be striped, encoded using a 10+4 Reed-Solomon code and stored over 70 nodes. The final resource calculation must include a safety margin for the rebuilding of redundancy blocks after hardware faults.

6.10.3 CERN EOS system - Object storage software

The CERN developed disk-based storage system is using the following software technologies: CEPH, RADOS and XRootD.

A multi-petabyte storage system must be highly scalable with no single point of failure. CEPH is a free software storage platform meeting such requirements; it is designed to present object, block, and file storage from a single distributed computing cluster. The object storage component of CEPH is called RADOS (Reliable Autonomic Distributed Object Store).

CEPH/RADOS

RADOS is implemented by a network of two types of daemons:

- cluster monitor daemons (MON) keep track of the current cluster configuration, state and node failures. They use the PAXOS protocol to elect the currently active monitor collecting and gossiping cluster state and there should be at least three of them in a cluster. Odd numbers are preferred.
- object storage daemons (OSD) store the actual objects and their metadata on different storage backends (filestore, key-value store, memory-store) using replication or erasure encoding to provide fault tolerance. One OSD runs on each storage device.

Reliability, data availability & storage overhead

In a capacity driven storage system the overhead for reliability has to be kept at a minimum level for a minimal cost.

Today CEPH offers two erasure code plug-in libraries (Jerasure, Intel ISA-L) providing Reed-Solomon encoding of objects. Reed-Solomon encoding splits each object into K pieces and creates M parities. The K+M fragments are stored on K+M OSDs on distinct failures domains (typically distinct hosts).

Large storage systems like GoogleFS use (K=9, M=3), Facebook HDFS uses (K=10, M=4) ensuring no data loss for up to three or four concurrent disk failures in each group of 12 or 14 OSDs. A large choice of K can reduce further the storage overhead. The price to pay is higher repair traffic in case of a disk failure as the reconstruction of a disk of size V means reading a volume of K*V.

In the CERN Computing Centre about 2.5 % of disks fail in a year. This represents an average of 1 disk per week for every 2.000 disks or (assuming 10 TB disks) or one disk per week in 20 PB of raw storage and an average repair traffic of 264 MB/s or ∼16 MB/s per disk drive in a failure group.

A possible configuration for a large ALICE storage pool could be (K=16, M=3) with 18.75 % storage volume overhead or (K=20, M=4) with 20 % storage volume overhead.

Erasure encoding CPU consumption

Reed-Solomon encoding requires Galois-Field multiplications which are accelerated on modern CPUs using AVX extensions. A Xeon 2.2 GHz core CPU can encode ∼5 GB/s per core which will not be the limiting factor of the OSD setup. The additional CPU time requirements are negligible in a large storage installation in case OSDs run on the computing nodes.

OSD CPU consumption & memory requirements

The OSD daemon is not lightweight in terms of CPU and memory consumption. Each OSD should be matched by one core and 2 GB of memory in the storage cluster. Under normal load scenarios the daemon does not require 2 GB of memory but for reconstruction and storage peering, it may do so.

CEPH deployment model

There are three possible deployment models:

- Storage and computing clusters are separated: this is the worst scenario in terms of cost since more nodes and network ports are required.
- Storage and computing clusters are overlaid: each computing node contributes with attached storage devices to the storage pool. There is no data locality: data are read from remote machines and the network design has to take into account the cross node traffic.
- Embedded storage server and separate computing cluster: Seagate is developing embedded storage servers on shingled disks. The idea is to run the CEPH OSD on an ARM processor on each disk. In this model all disks are attached to a 1 Gb/s port and are accessed from a separate batch cluster.

CERN IT has been evaluating CEPH since early 2013 and has been running a 3 PB production cluster since November 2013.

CEPH can be used as a low-level storage platform to provide a robust multi-petabyte storage platform. Using erasure encoding, the storage system is highly reliable with ∼20% storage space overhead. Several high-level APIs and storage services allow the efficient use of the object storage layer for application and physics data storage and processing. At the present time, the most cost effective storage deployment is an overlay deployment of storage and computing: computing nodes with disk arrays attached.

6.10.4 Reliability and Recovery

The reliability of hard disks is becoming an important issue for large installations as the rebuild of volumes after a disk failure will take very long time. With typical Mean Time Between Failures (MTBF) values of for enterprise disks and for cold-storage disks, a pool of 30 PB will suffer from one disk failure every 6 days. With double parity calculation, the build speed observed is in the order of 5 TB/day (per controller in a disk array). This means that e.g. a 80 TB volume consisting of 10 disks of 8 TB will take 16 days to recover from a single disk failure. But in the meantime, another 8 disk failures might occur in the system naturally, plus additional ones on the disks used for the rebuild which are under high stress. Not only will this cause a significant slow-down of the ongoing rebuild but also put the data at risk as another (3rd) disk failure will break the volume. Once the first rebuild has finished, a second one will then restore the initial double-parity. It is therefore important to use enterprise-grade disks with high MTBF values in order not to suffer from too many rebuild cycles. In addition, higher parity levels are needed, as with long rebuild times the risk of losing more than two drives per volume is no longer acceptable. Therefore, the classic RAID systems are no longer an option and will be replaced by DDP or other systems such as CEPH which provide higher parity levels.

Another issue for data integrity are disk-related failures: lost writes, lost reads and off-track writes. These errors are rare and typically appear at a rate of 1/year/PB, so for ALICE this would mean one un-recoverable error every ten days for 30 PB. Usually these errors are not detected by RAID or parity algorithms because successful reads are handled as pass-through (i.e. not checked against parity). In any case, under these conditions a RAID or parity algorithm cannot restore the original data, so it is up to the application to detect and reject the data. These types of errors affect at least one file, off-track writes rather two files or the partition table (in this case a file system intervention is needed).

The O^2 system will need a protection against this type of failure, either at the level of the file system or in the application software.

6.11 Control, configuration and monitoring

Several tools providing essential CCM functions have been identified and are listed in Tab. 6.8. As new tools are expected before the CCM is implemented, the final selection will be deferred until the

opportune moment. Nevertheless, several tests have been conducted to ensure that critical performance requirements can be fulfilled with existing tools.

Table 6.8: Tools to implement CCM functions.

Module	Function	Tools
All	Inter Process Communication	DIM, ZeroMQ
Control	Start/stop processes	DDS
Control	Send commands to processes	SMI, ZeroMQ
Control	Task Management	SMI
Control	State Machine	SMI, Boost Meta State Machine
Control	Automation	SMI
Configuration	System Configuration Management	Puppet, Chef
Configuration	Configuration Distribution	ZooKeeper
Configuration	Dynamic Process Configuration	ZooKeeper
Monitoring	Data Collection and Archival	MonALISA, Zabbix
Monitoring	Alarms and Action Triggering	MonALISA, Zabbix

6.11.1 DIM and SMI

The current ALICE Control system uses the software stack DIM/SMI for Inter Process communication, state machine definition, command distribution and process synchronisation. As there will be a large increase in the number of processes and commands for the O^2 system, performance tests have been performed with DIM/SMI to assess its potential for the future Control system.

As shown in Fig. 6.18, the initialisation time is the limiting factor. Once this initial STANDBY state is reached by all processes, subsequent commands are executed almost instantaneously. Given the current performance of DIM/SMI, the maximum number of processes compatible with a reasonable startup time (less than 5 minutes for a complete restart) is roughly 25000, which is lower than the estimated number of processes for the O^2 system.

To control the estimated number of processes, the CCM system could include logical objects that represent the state of multiple processes on a single node, thus greatly reducing the number of processes that must directly use SMI.

6.11.2 ZeroMQ and Boost Meta State Machine

A small message-broadcasting system was developed to test the performance of ZeroMQ to send control messages and the Boost Meta State Machine library for state machine definition.

As shown in Fig. 6.19, 67500 processes could be started and initialized in less than 1 minute. Once all processes are initialized, subsequent commands are executed almost immediately.

This approach constitutes an alternative to SMI, allowing for all the estimated number of processes to use a similar mechanism for Inter Process communication, state machine definition, command distribution and process synchronisation.

6.11.3 MonALISA

In Runs 1 and 2, MonALISA is used for monitoring all Grid processes and activities. The service layout follows closely the resource distribution with at least one data collecting service in each computing centre. This first layer of services implements data aggregation and filtering in order to generate high-level views from the $\sim$7 million distinct parameters that are published by the $\sim$60000 concurrently

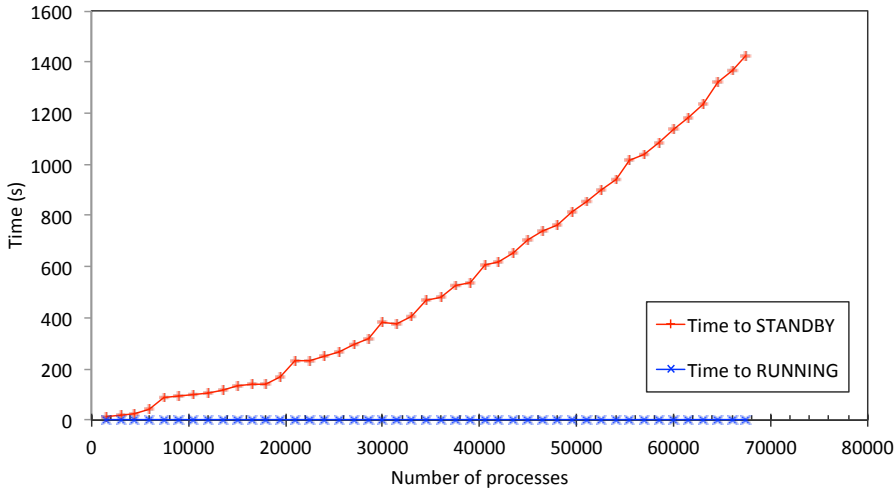

Figure 6.18: SMI performance measurements as a function of the number of processes. The red line shows the time it takes for all processes to reach the initial STANDBY state. The blue line shows the time it takes for all processes to move from STANDBY to the final RUNNING state (STANDBY - CONFIGURING - CONFIGURED - STARTING - RUNNING).

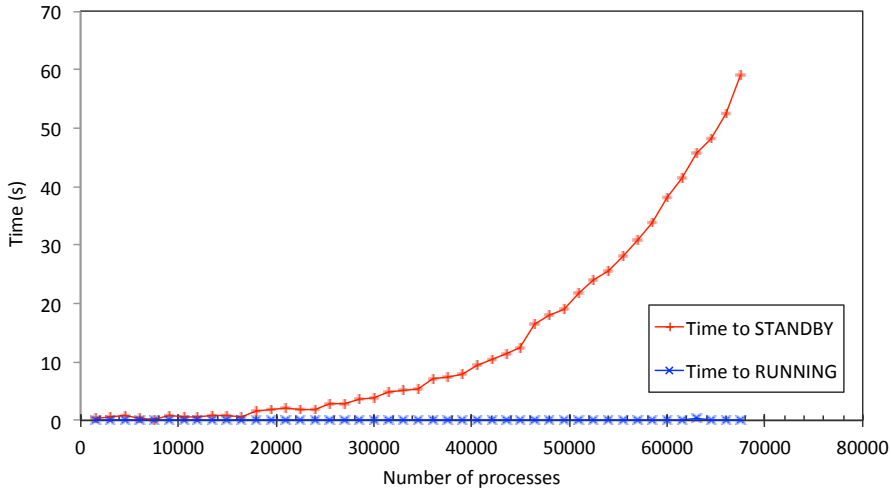

Figure 6.19: ZeroMQ and Boost Meta State Machine performance plot measured as a function of the number of processes. The red line shows the time it takes to start all processes and for each individual process to reach the initial STANDBY state. The blue line shows the time it takes for all processes to move from STANDBY to the final RUNNING state (STANDBY - CONFIGURED - RUNNING).

running processes. While the entire system matches in scale the expected rates for the O^2 system, it has the advantage of distributing this load on 100 different services.

To evaluate the requirements for the O^2 monitoring system, a setup with one collecting machine and 12 sender hosts with 1000 threads each was deployed. Each thread sends a User Datagram Protocol (UDP) packet with a single parameter at fixed intervals. In Fig. 6.20 the received message rate and the lost message rate can be seen.

This setup was thus able to collect up to 50 kHz of messages before the loss became significant. As such, up to 15 collecting services would be needed to handle the largest expected monitoring data rates from

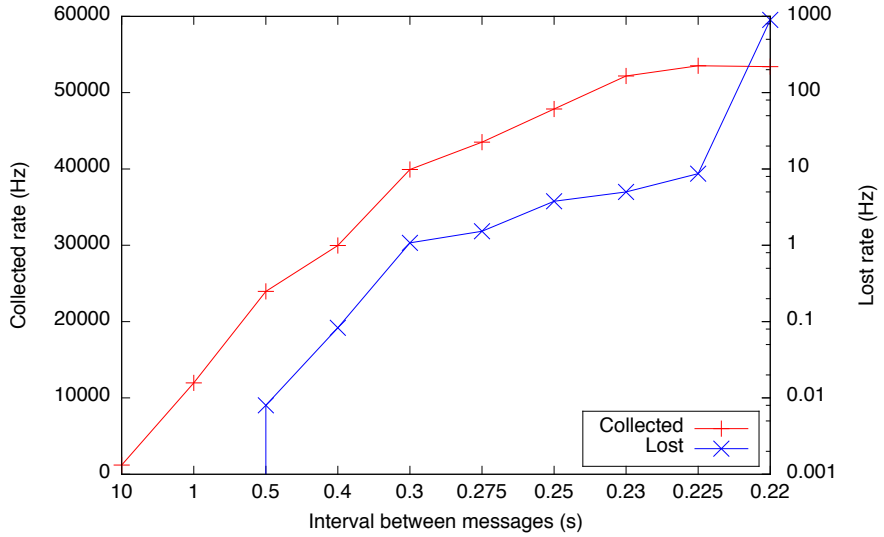

Figure 6.20: Collected and lost message rates in MonALISA as a function of sleep between messages.

the O^2 system.

6.11.4 Zabbix

The Zabbix monitoring system is currently used in the ALICE HLT and DAQ for computing cluster management. The O^2 system requires a monitoring tool capable of handling very high rates of incoming data in a scalable and efficient way. Several performance tests have been made to determine Zabbix abilities to cope with future monitoring requirements.

Fig. 6.21 shows the measured Zabbix performance. As seen in the graphic, Zabbix is able to handle up to 25 khz without major performance issues. Higher rates decrease the percentage of received values which could be solved using multiple monitoring servers.

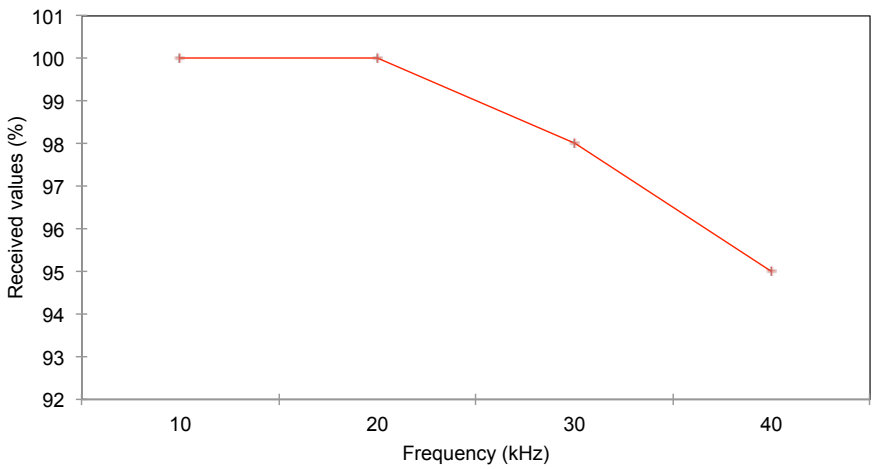

Figure 6.21: Zabbix performance measurements as a function of the values frequency.

Chapter 7

O^2 software design

7.1 Introduction

The O^2 software consists of a framework which runs on the O^2 farm as well as on the other sites (T0, T1, T2, see Chap. 4). It has been designed with generic functionalities that can be specialised or replaced. The framework also drives the execution of the programmes and implements the general design principles presented in this introduction and in Sec. 7.2.

The effect of advances in technology and frequency scaling on modern applications is not as great as in the past. Now, the primary method of gaining extra performance is by parallelisation of the application code and by using multi- and many-core CPU capabilities as well as introducing specialised hardware accelerators. Dealing with such systems is usually complex and error prone, especially for non-experts. For this reason, the O^2 framework aims to minimise the development effort for end users while running transparently in a distributed and heterogeneous environment.

As shown in Fig. 7.1, the O^2 software not only relies on general libraries and tools such as Boost [1], ROOT [2] or CMake [3], but also on 2 other frameworks called ALFA and FairRoot. ALFA is the result of a common effort of the ALICE and FAIR experiments to provide the underlying communication layer as well as the common parts for a multi-process system. The multi-process approach, rather than a purely multi-threaded one, is justified by the need for high-throughput parallelisation and the necessity to have a flexible and easy-to-use software framework. At the same time the system is fully capable of multi-threading within processes when required.

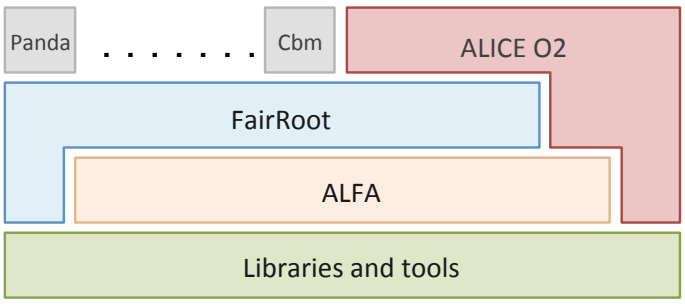

Figure 7.1: O^2 software ecosystem (not drawn to scale).

This chapter is organised in sections describing the main components of the O^2 software. The future packages of the software will follow a similar structure. The order in which the packages appear in this chapter is from lower to higher level functionalities.

7.2 ALFA

ALFA is the new ALICE-FAIR concurrency framework of the O^2 software system for high quality parallel data processing and reconstruction on heterogeneous computing systems. It provides a data transport layer and the capability to coordinate multiple data processing components. The existing algorithms of these components must be optimised for speed and the framework has to manage the high throughput of data between these algorithms. Moreover, the modules, these algorithms and associated calibration and reconstruction procedures, must run very efficiently on a highly parallel system, with several levels of parallelism and granularity. This will require the code to be developed and optimised taking into account newly emerging parallel architectures.

ALFA is a flexible, elastic system which balances reliability and ease of development with performance by using multi-processing in addition to multi-threading. With multi-processing, each process assumes limited communication and reliance on other processes. Such applications are much easier to scale horizontally to meet computing and throughput demands (by creating new instances) than applications that exclusively rely on multiple threads which can only scale vertically. Moreover, such a system can be extended with different hardware (accelerators) and possibly with different or new languages, without rewriting the whole system.

The modules of the ALFA framework and the libraries and tools it uses are show in Fig. 7.2

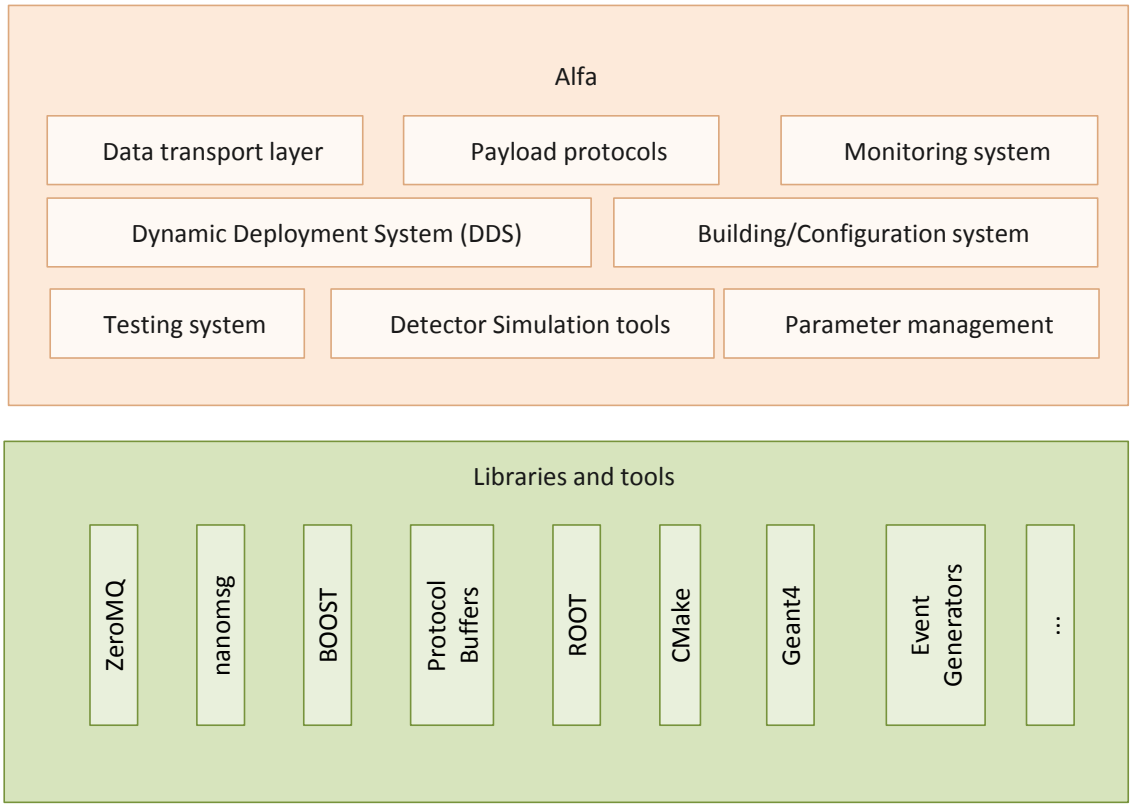

Figure 7.2: Modules of the ALFA framework and the libraries and tools used.

7.2.1 Data transport layer

The data transport layer is the part of the software which ensures the reliable arrival of messages and provides error checking mechanisms and data flow controls. The data transport layer in ALFA provides

a number of components that can be connected to each other in order to construct a processing topology. They all share a common base called *device*. Devices are grouped in three categories:

- Source: Devices without inputs are categorised as sources. A sampler is used to feed the pipeline (Task topology) with data from files.
- Message-based Processor: Devices that operate on messages without interpreting their content.
- Content-based Processor: This is the place where the message content is accessed and the user algorithms process the data.

The data transport layer is based on ZeroMQ [4], a very lightweight messaging system, specially designed for high throughput and low latency scenarios. ZeroMQ is an open source, embeddable socket library that redefines the term socket as a general transport endpoint for atomic messages. ZeroMQ sockets provide efficient transport options for inter-thread, inter-process and inter-node communication. Moreover it provides a Pragmatic General Multicast (PGM), which is a reliable multicast transport protocol.

A multi-part message is a message that has more than one frame but is sent as a single one on-the-wire. The multi-part message support in ZeroMQ allows the concatenation of multiple messages into a single message without copying: all parts of the message are treated as a single atomic unit of transfer, while strictly preserving the boundaries between message parts. Such features are crucial for implementing Multiple Data Headers (MDH see Sec. 7.5.2).

7.2.2 Payload protocol

ALFA does not dictate any application protocols. Potentially, any content-based processor or any source can change the application protocol. Therefore, only a generic message class is provided that works with any arbitrary and continuous chunk of memory. A pointer must be passed to the memory buffer, the size in bytes and, if required, a function pointer to the destructor, which will be called once the message object is discarded.

The framework supports different serialisation standards that allow for data exchange between different hardware and software languages. Moreover, some of these include a built-in schema evolution which naturally simplifies the development of the software and guarantees the backward compatibility of the payloads. ”Object serialisation” actually means writing the current values of the data members of an object to a socket or file. The most common way of doing this is to decompose the object into its data members and write them to disk or to socket. Only the persistent data members are written and not the methods of the class. To decompose the parent classes, the serialisation process should also be called for the parent classes. This moves up the inheritance tree until it reaches an ancestor without a parent. Many easy-to-use tools exist for this purpose. The following are supported by the framework:

- Boost serialization

  This method depends only on ANSI C++ facilities. Moreover, it exploits features of C++ such as RTTI (Run-Time Type Information), templates or multiple inheritance. It also provides independent versioning for each class definition. This means that when a class definition changes, older files can still be imported to the new version of the class. Another useful feature is the save and restore of deep pointers.

- Protocol buffers

  Protocol buffers [5] are Google’s language and platform independent mechanism for serialising structured data. The structure of the data is defined once and used to generate code to read and

write data easily to and from a variety of data streams, using a variety of languages: Java, C++ or Python.

– ROOT

The ROOT Streamer can decompose ROOT objects into data members and write them to a buffer. This buffer can be written to a socket for sending over the network or to a file.

– User defined

In case it is decided not to use any of the above methods, binary structures or arrays can still be written or sent to a buffer. Although this method does not include any overhead for size of the data, issues can occur and will need to be managed. These include: schema evolution, different hardware, different languages.

7.3 Facility control, configuration and monitoring

The CCM components coordinate all the O^2 processes, ensuring that both the application and environment parameters are properly set. They also gather information from the O^2 system with the aim of identifying unusual patterns and raising alarms. The CCM systems interface with the Trigger, the DCS, the Grid and the LHC to send commands, transmit configuration parameters, submit jobs and receive status and monitoring data.

7.3.1 Tasks

The O^2 functions are defined as sequences of actions known as *tasks* which can be *base* or *composite*. *Base tasks* are sequences of precise instructions executed on a single node. *Composite tasks* are groups of *base tasks* and/or *composite tasks* that can be executed on a single or multiple nodes in sequence or concurrently (see Fig. 7.3). The Control system must also ensure that *tasks* are executed in the correct order and that *task* execution failures are properly handled. Examples of *tasks* are the reset of detector FEE using the read-out link or the execution of Monte Carlo simulations on unused resources. *Tasks* should be considered as an input to the Control System and their definition handled via proper editing and validation tools.

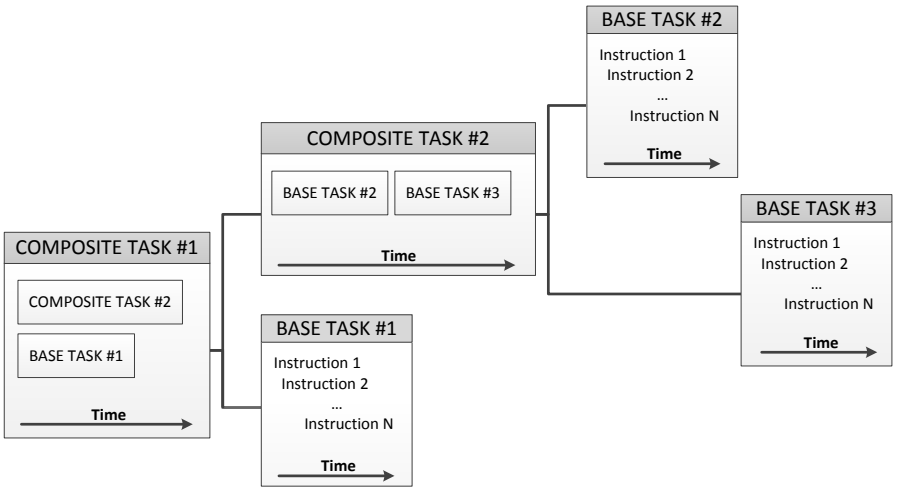

Figure 7.3: Definition of an abstract task.

7.3.2 Process state machine

To ensure uniformity in the control of the O^2 processes, a base state machine is defined as presented in Fig. 7.4. Each O^2 process must implement it as a prerequisite and can extend it to fulfil specific requirements. This state machine foresees dynamic reconfiguration, allowing for faster configuration changes and greater flexibility. It also foresees a pause and resume mechanism to allow process throttling according to available resources and operational priorities.

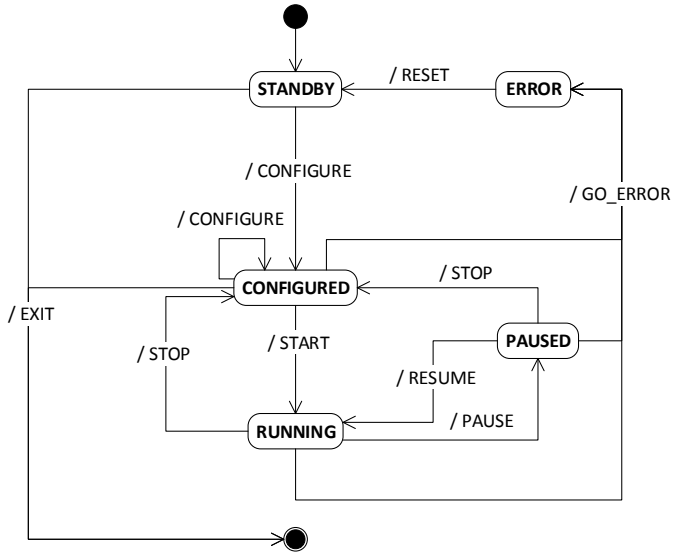

Figure 7.4: Process base state machine.

7.3.3 Roles and activities

The control and operation of the O^2 system is based on logical entities called ”Roles” that represent and group computing nodes and functional blocks. Each role has specific commands, monitoring metrics and configuration parameters. As shown in Fig. 7.5, the roles are organised in a hierarchical structure representing the ALICE detectors and the O^2 farm. This hierarchy provides not only a closer match to the experiment’s setup (e.g. each individual detector is read out via a set of specific non-interchangeable FLPs) but also increases scalability and parallelism by reducing the scope of each role to a subset of the whole system.

Activity

ALICE operations are made up of different tasks which are limited in time and normally do not make use of the full O^2 farm. To achieve an optimal usage of the available resources, it is possible to execute multiple tasks in parallel. A task executed by a set of roles during a finite time period is called ”Activity” and includes both synchronous and asynchronous tasks as defined in Sec. 5.1. Activities are identified by an Activity Number for bookkeeping purposes. Examples of Activities are: physics data taking; detector calibration; reconstruction passes.

Run

A ”Run” is a finite data taking time period with stable experimental conditions represented by a unique identifier called the ”Run Number”. It corresponds to a specific data set generated at the end of the synchronous part of the data flow. To increase operational flexibility, a physics data taking Activity

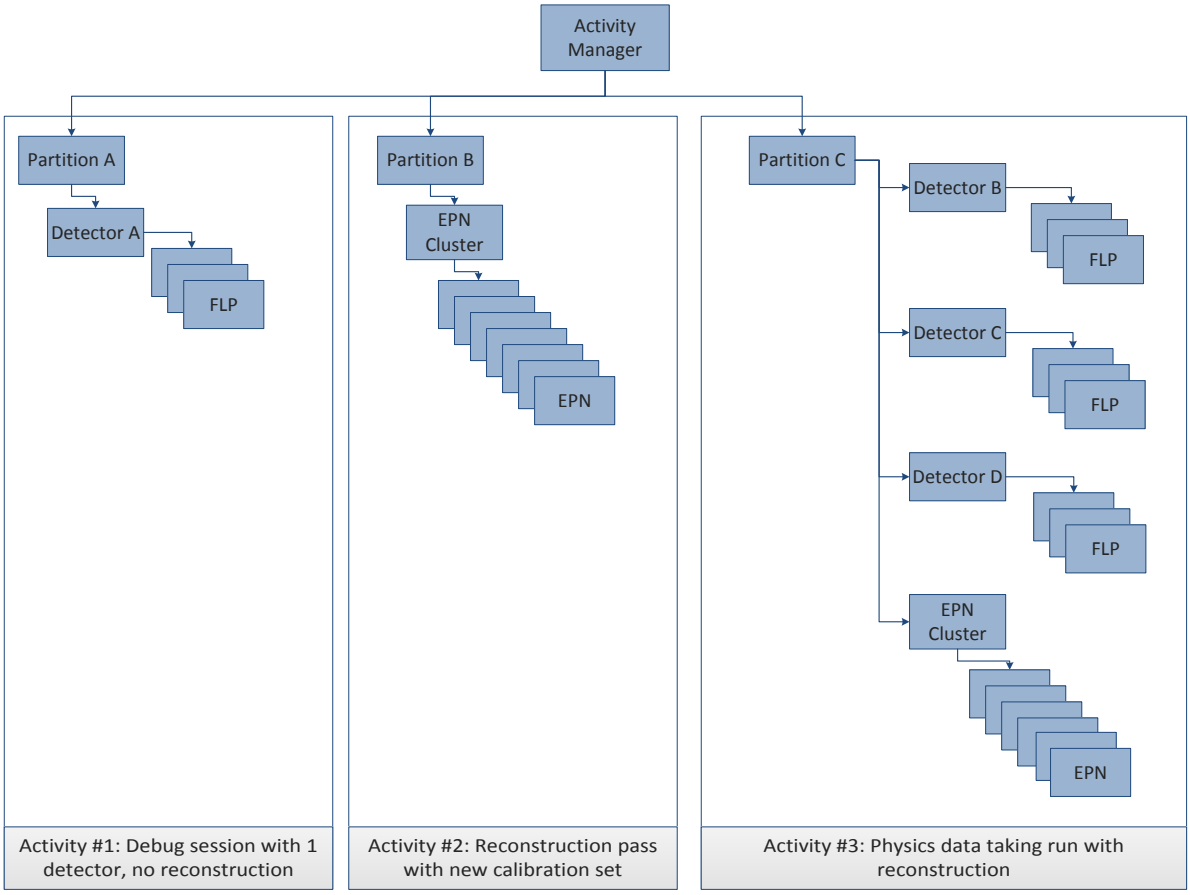

Figure 7.5: Examples of parallel Activities and their associated Partitions.

can include multiple Runs thus allowing for fast run transitions, whenever the experimental conditions change. Run transition tasks dynamically handle the allocation of resources and the required reconfiguration of components, therefore ensuring that unnecessary operations are skipped.

Specific roles

The FLP role represents the functional blocks of a single ALICE detector executed on a single FLP node. Typical actions include: starting/stopping data input/reduction processes; monitoring data rates; sending commands to detector FEE; enabling/disabling read-out links. Multiple FLP roles can coexist on the same FLP node.

The EPN role represents the functional blocks of a specific Activity on a single EPN node. Typical actions include: starting/stopping data reduction/reconstruction processes; monitoring reconstruction metrics; configuring reconstruction data sources. Multiple EPN roles can coexist on the same EPN node. EPN roles can be dynamically assigned to, or removed from, an ongoing Activity.

The Detector role represents an ALICE detector and groups FLP roles defined on FLP nodes physically connected to a specific detector. Typical actions include aggregation of monitoring values per detector and send commands and configuration parameters to all its FLPs.

The EPN Cluster role groups all EPN roles participating in a specific Activity. Typical actions of this role include aggregation of monitoring values and send commands and configuration parameters to all the EPNs in the cluster.

The Partition role is the root node of the role hierarchy responsible for the execution of a specific Activity that includes Detector and/or EPN Cluster roles.

Activity manager

The Activity Manager is responsible for the instantiation and termination of Activities, including the creation and destruction of the Partition hierarchies and sending commands to the Partition roles. Users interact with the Activity Manager to control and configure Activities.

Roles locking mechanism

To avoid conflicts in the usage of available resources (e.g. read-out links) and to simplify the management of parallel Activities, a role locking mechanism guarantees that each role can only be assigned to a single Activity. All the same, to increase operational flexibility and to optimise the use of resources, multiple roles can be defined in a single node and assigned to concurrent Activities.

7.3.4 Agents

Each defined role is implemented by a logical entity called ”CCM Agent”, consists of several processes and/or software modules. As seen in Figure 7.6, a CCM Agent starts, stops and sends commands to local O^2 processes, configures and monitors them and interacts with other CCM Agents. The CCM Agents implementing the Partition roles also interact with the Activity Manager.

Control

The CCM Agent receives a *task* to execute from the parent CCM Agent (or in the case of Partition CCM Agents, from the Activity Manager). Based on the *task* content, it launches and sends commands to the required local O^2 processes. Once these processes are no longer needed, they are either stopped or put in an idle state by the CCM Agent. The CCM Agent has also the role of forwarding the required subtasks to descendant CCM Agents and handling asynchronous events based on the monitoring data (e.g. a dead process that needs to be restarted).

Configuration

The Partition CCM Agents retrieve the configuration of the Activity from a central repository where all configuration parameters are stored. To minimise network traffic, the other CCM Agents only receive from the parent CCM Agents the configuration subset for itself and for its descendant CCM Agents. The configuration of the local O^2 processes is performed as part of the *task* sequence executed by the CCM Agent. The descendant configuration parameters are forwarded to the corresponding CCM Agents. The configuration distribution and usage is optimised by keeping a local copy of previously used configurations.

To allow for fast reconfiguration, a CONFIGURE command is sent only to the O^2 processes which are affected by a given configuration change. For example, to remove a malfunctioning GBT from data taking, the following sequence of commands is executed: STOP on all O^2 processes, CONFIGURE on all concerned O^2 processes and START on all O^2 processes.

Monitoring

All main processes are instrumented to send periodic, high frequency heartbeats (not to be confused with heartbeat triggers) that are critical in establishing the functional status of the system in general. In addition, the main processes have access to an API to explicitly publish internal parameters to be monitored as events or as periodic metrics.

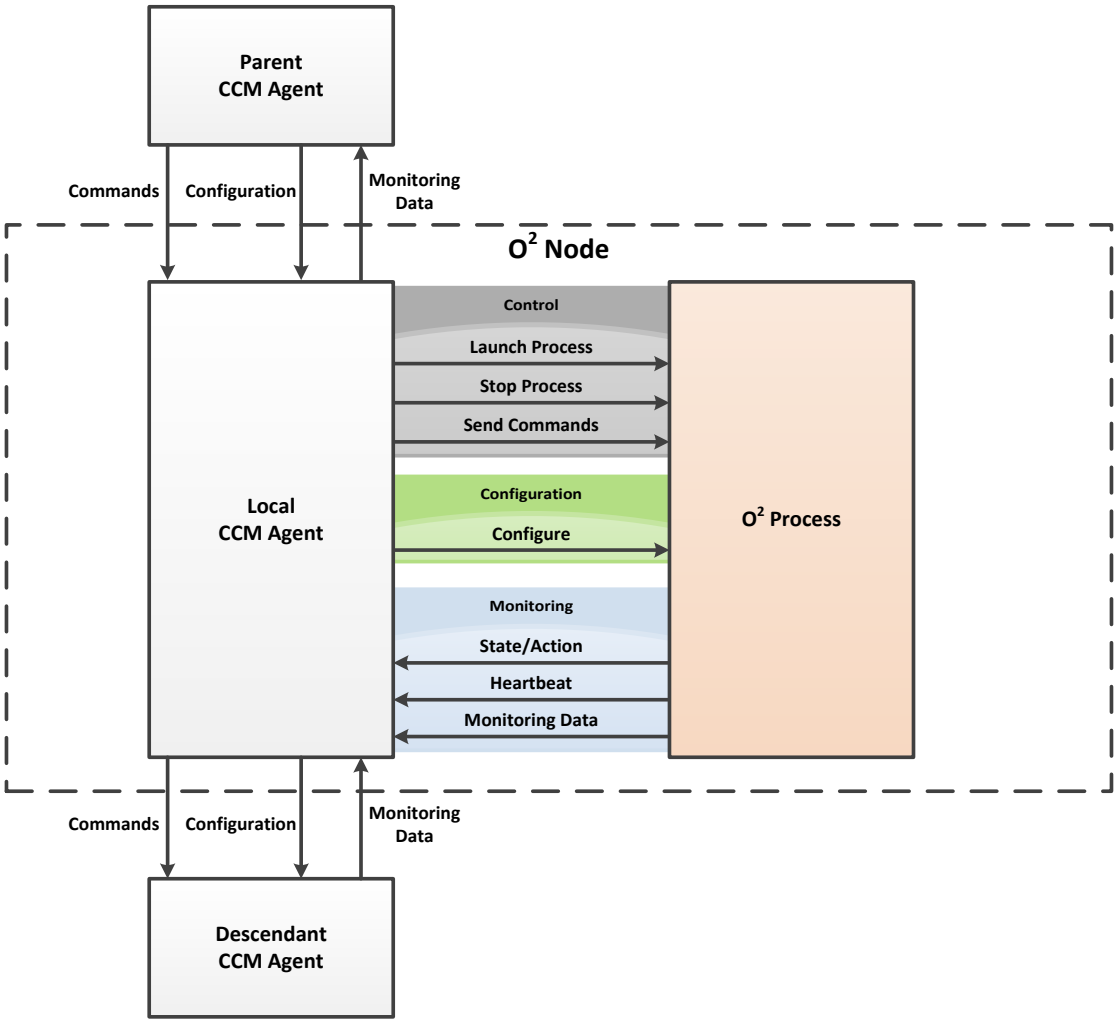

Figure 7.6: Interaction between CCM Agents and Processes.

Processes are also monitored from an operating system point of view, periodically reporting the memory footprint, CPU consumption, number of threads, file descriptors and other relevant metrics. This can be done either internally, from the library providing the monitoring API, or externally by the CCM agent that started the processes.

The CCM agents also implement monitoring data aggregation to help reduce the volume of data received continuously by the subscribers, in particular by the long term archival clients. At this level they can also trigger events and in particular raise alarms under certain conditions, for example: processes not running; hardware failing; backlog over certain thresholds. The actual configuration of the triggers and alarms is distributed by the Configuration component of the CCM system and can be applied dynamically on the running system.

7.3.5 System monitoring

This monitoring of all the components of the O^2 system assesses the status and health of all hardware and software entities. It also provides high-level views of the global system and archives relevant metrics for long-term analysis and forensic investigation.

A monitoring API allows any software component to publish heartbeat and explicit monitoring data to a

common data store. The same data store also gives periodic reporting of operating system views of the main processes and other critical services as well as monitoring data collected from the infrastructure. This includes: server health; utilisation and fabric monitoring data like switches, routers, power supplies and other intelligent components.

Other external components to the CCM can also push monitoring data into the same data store, or, if the implementation does not allow it, dedicated modules will pull the metrics from the external systems and inject them into the common data store.

The monitoring API also allows current monitoring values or the history data to be queried. History data are only available for the archived metrics. In addition, this API subscribes to arbitrary cuts in the common monitoring metrics namespace so that it can be notified in near real time of newly available data for those metrics.

This mechanism will be used by the Control system to assess the health of the system in general and trigger actions accordingly, including when processes fail to report as running or critical services report an error condition.

The monitoring system uses two classes of messages: either regular monitoring data stream that is consumed by the subscribers or simply discarded and out of band events that are persistently stored until they are acknowledged.

7.4 Readout

The *readout* process will be running in all the FLPs participating in data taking. Its main objective is to move data from the detector electronics into the memory of the hosting PC as fast as possible. The general *readout* structure contains a few routines needed to initialise the read-out electronics used to collect data and a main loop where data are moved inside the memory of the PC. A general schema of the read-out process is shown below.

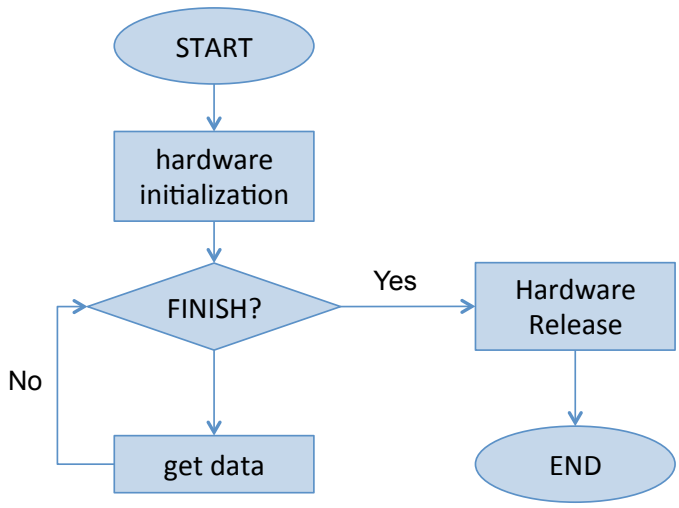

Figure 7.7: Main phases executed by read-out

Readout has three main phases:

Start of run phases

At each start of run *readout* performs the following procedures:

– Initialisation of the read-out hardware installed in the FLP needed to collect the data,

– Initialisation of the communication with different EPNs used during the run.

Main control loop

Each time an event is received, *readout* performs general data quality checks and in case of FATAL errors it can request to stop the data taking. Inside this loop, *readout* updates different counters, like the run number and the number of bytes received. The statistics of each run can be used for later studies. The loop continues until the end of run is requested.

End of run phases

When the end of run is requested the following operation is executed:

– Release of the hardware channels used during data taking.

While the main structure of *readout* is independent from the protocol used to collect data, the communication with the read-out board is affected by the data transmission protocol and the related hardware used. The read-out program will be tailored to read different read-out boards: the CRU and the C-RORC.

Readout needs drivers and APIs to access the hardware. The CRU is a custom board and will required dedicated firmware and driver to transfer data inside the PC memory. However, the FPGAs considered for the CRU include the PCIe interface with a DMA engine.

7.5 Data model

An important task of FLPs from the data model perspective is the creation of the metadata to facilitate navigation within Time Frames. The structure of this metadata is described in Sec. 7.5.2. The EPN nodes collect the Time Frames produced by each individual FLP for a given time interval, then aggregate them in a folder-like structure which can be navigated using a Time Frame descriptor as described in Sec. 7.5.5. The reconstruction performed on EPN nodes will use these Time Frames as input, producing a lossless intermediate persistent format as output, migrated to the local storage. Subsequent calibration and reconstruction processing can run asynchronously with respect to the data taking flow to further reprocess this data.

While the most suitable format for pipeline processing and data transfers are raw C-like structures, the reconstruction, calibration or analysis/QA algorithms are usually accustomed to object-like APIs. Since performance is a critical aspect, the approach of transient objectification on top of flat C-like structures is adopted, using references or transient pointers. Using references is necessary to avoid or to minimise data duplication. The best strategy for this is to keep the data in the original buffers and reference them from objects with a higher level of abstraction, which also add metadata information as needed. This approach allows for C++ object-like manipulation of data while keeping the memory management under the control of the framework. Having a controlled contiguous memory allocation pattern for the most used data structures (e.g. clusters) is important for optimising both the memory layout and exploiting vectorised access in tight loops.

In the sections below, self-contained data blocks coming from the FEE are referred to as *raw data samples*. DAQ terminology is used for properties such as *attributes* (entities specific to each type of data) and *flags* (status and error fields common to all data samples). The concept of embedded commands is also used. These are data fields inside the trigger messages which are used to perform specific operations on the detectors and/or on the data processing nodes including: start data taking; calibrate; reset or

open/close Time Frame. The raw data samples coming from the FEE can be associated with different procedures such as read-out of physics or test data, collection of detector-specific counters and flags, information associated to detector control.

7.5.1 Single data headers

The FLPs receive the raw data sample payloads, which can be either triggered or created by continuous read-out procedures, from all the FEEs which are assigned to them. The raw payloads for all read-out schemes will be individually preceded by a Single Data Header (SDH), created by the FEEs, describing the raw data itself as seen by the source equipment. Each SDH will be followed by a Single Data Block (SDB) which will contain the payload created by the FEE and the format of which will be FEE-dependent, as described in Fig. 7.8.

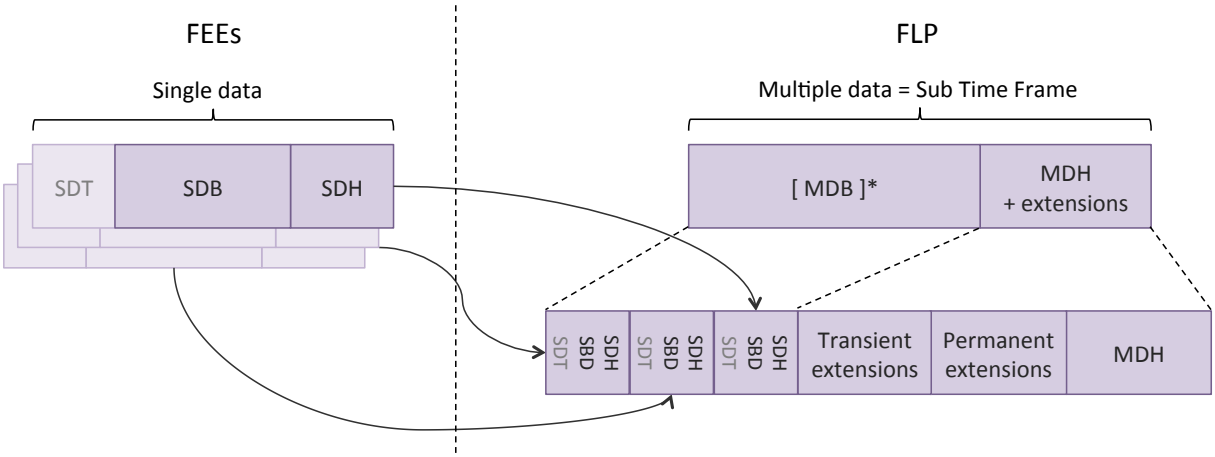

Figure 7.8: Data format at the level of the FLPs. [entity]* means zero or more occurrences of *entity*.

The SDH contains all the attributes common to all FEEs, including: precise time-stamping of the data block (either the time of a trigger or the beginning timestamp of a continuous read-out sample); status and error flags; unique FEE identifiers; SDH specific version ID. Some of these fields are mandatory while others are optional. In the latter case, a field-dependent NULL value must be used by the source FEE.

Whenever the FEEs need to tag the end of a SDB, a Single Data Trailer (SDT) has to be appended to it. The SDT, created by the FEE, will contain any information that cannot be sent within the SDH such as status and error bits associated to the end of the raw data read-out procedure and time-stamping of the end of the data block. The SDT presence is optional and declared by a dedicated attribute bit in the SDH prepended to the same data block. Although the fields making up a SDT and a SDH are not identical, each one has to use a common scheme: time-stamping, attributes, error and status bits.

One of the fields of the SDH is the data type. This is used to characterise the information stored in the SDB. So far, three data types coming from the FEEs have been identified:

- PHYSICS (PHY): physics data, either triggered or continuously read out.
- CALIBRATION (CAL): detector specific calibration data, always triggered.
- HEARTBEAT EVENT (HBE): dedicated triggered events used to re-synchronise the FEEs and to send them commands.

More data types like information associated to Detector control procedures, can be declared if necessary.

The FLPs have also to create and replace payloads locally and autonomously. The newly created data blocks are described by the same SDH (and optionally SDT) as if they were coming from the FEE. The fields of the FLP created SDHs are usually filled using data (SDHs, SDBs, and SDTs) coming from the FEEs. Specific data types must be allocated according to the content of the payloads.

All payloads may contain within them payload specific headers and/or trailers. The specifications of these data blocks, like format and positioning in the payload, are not defined here as the actual implementation will be payload specific. All SDHs will include a unique ID that gives the source of the payload, either FEE, in which case a Detector-specific ID will be used, or a software procedure running on the O^2 farm.

7.5.2 Multiple data headers

The FLPs encapsulate the data blocks coming from the FEEs using Multiple Data Headers (MDHs), as well as the data blocks generated by the FLP processing. An MDH fully describes one Multiple Data Block (MDB), which contains zero or more sets of SDH+SDB(+SDT). FLPs handling multiple FEEs are expected to encapsulate multiple triggered payloads (one per FEE) using a single MDH. Continuous read-out payloads, on the other hand, will be encapsulated individually. The MDHs contain a summary of the payload(s) plus some information collected or generated by the FLPs themselves: MDH version ID; FLP ID; data type; number of payloads. Each MDH is equivalent to what is known in trigger-driven data acquisition architectures as an *event* and has to serve two main purposes: the encapsulation of correlated information coming from different links; the provision of transparent access to all data blocks independent of their type.

In case the MDB block contains dynamic data, a Multiple Data Trailer (MDT) will be used to signal the end of the block.

Besides the specific device (data block sender) extensions, the FLPs may add more extensions to the MDHs dedicated to local processing procedures like a list of scattered memory pages used to store the payloads or pointers to the SDHs. These extensions, transient by nature, will be dropped when the data leave the FLPs for the EPNs.

7.5.3 Time Frame descriptor

The data granularity in Run 3 is driven by the so called *heartbeat* (HB) trigger signals which will be fired at equal time intervals, typically on the same bunch crossing ID. The spacing between two consecutive HBE is dictated mostly by the TPC drift time and is foreseen to have values in the range of O(10) milliseconds. The HBE will carry both the information for synchronising the HW/ SW components in the system and the steering commands to be propagated to all relevant processing components to enforce a given behaviour or functionality of the system.

For the data model, all input data blocks acquired in a HB interval and all new data produced based on those data blocks are assigned to a single Time Frame descriptor tag. This has the double advantage of easy navigation through the information within the HB interval in a folder-like fashion plus the eventual correlation of data coming from different streams. In addition, the Time Frame descriptor permits the pushing of relevant HB information and commands downstream to higher level processing units. This section describes how the Time Frame descriptor can be aggregated from MDH at the level of the FLP and how the navigation can be done on the EPN.

A schematic representation of the input data aggregation on each FLP node for both continuous read-out and triggered detectors is made in Fig. 7.9. The HB trigger chops the data in equidistant time frames. The FLP logically groups together all inputs or produced data within the same frame so that they can be shipped and further processed on the EPN nodes. The MDH is self-contained, providing pointers to the

transient data blocks in the FLP input buffer, including information about correlated events, as described in Sec. 7.5.2. The Time Frame descriptor, behaving like a folder for a given FLP corresponding to a unique frame ID, summarises all MDHs created transiently for a given Time Frame interval.

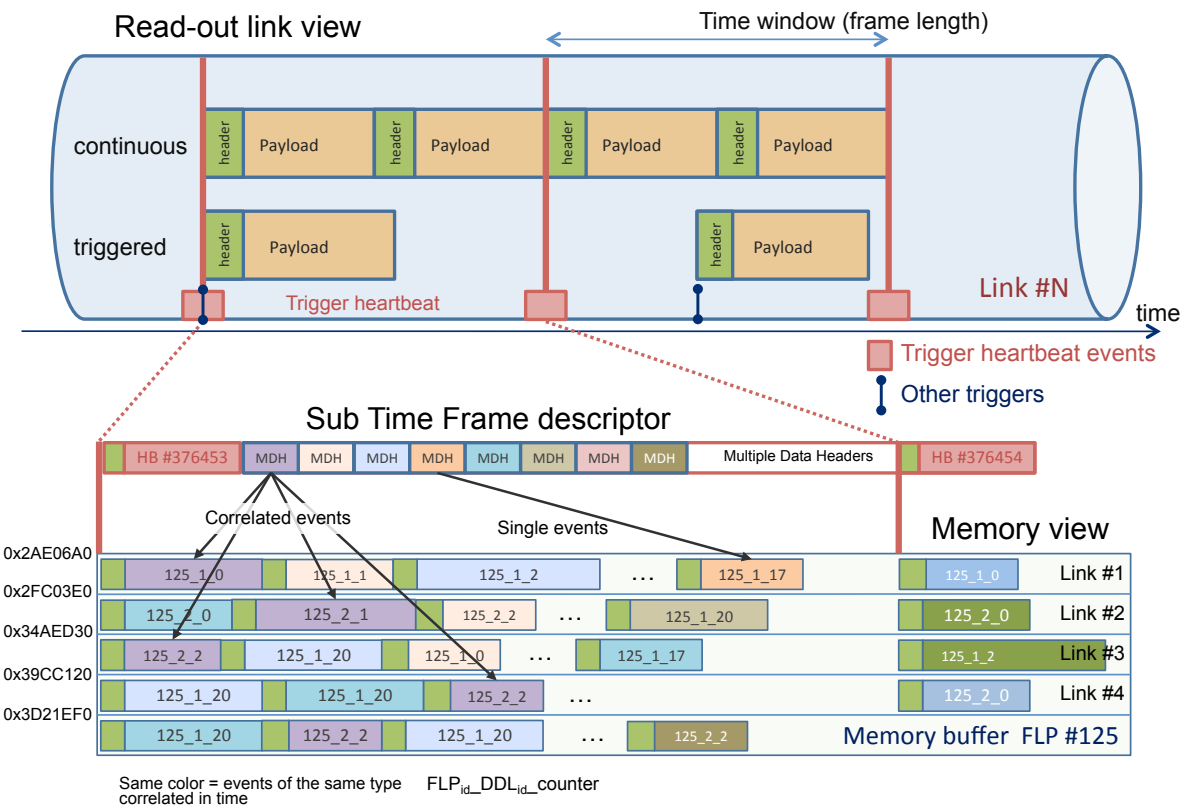

Figure 7.9: The Time Frame descriptor assembled on FLPs. Top: schematic flow of data blocks (events) as sent by the FEE. Bottom: The aggregation of individual data blocks into MDH headers.

According to the data flow approach, in a scenario where the MDH + MDB (+ MDT) are being scheduled for asynchronous dispatching as soon as they are assembled, the full Time Frame descriptor can only be aggregated after the end frame HBE and sent to the EPN as a trailer in the frame data stream. Note that all absolute addresses used to navigate through transient data blocks while on the FLP become relative addresses to the current frame stream offset and will eventually be reconverted to absolute addresses after landing on the EPN.

7.5.4 FLP data aggregation

Figure 7.8 shows the format of the data coming into the FLPs as generated by the FEEs and while stored inside the FLPs with the following considerations:

1. SDTs are always optional.
2. For continuous read-out detectors, if the single block is coming from a continuous read-out procedure, then there will be exactly one single data block per event.
3. For triggered events, coming either from triggered detectors or from continuous read-out detectors when these are triggered by the central trigger system, the expected event structure is one single data block for each channel belonging to any of the detectors that have been triggered.

The data blocks are grouped by the FLPs in Time Frames. Each Time Frame is delimited by two HBEs (a "Start HBE" event and an "End HBE" event) before being sent to one specific EPN. These two HBEs will

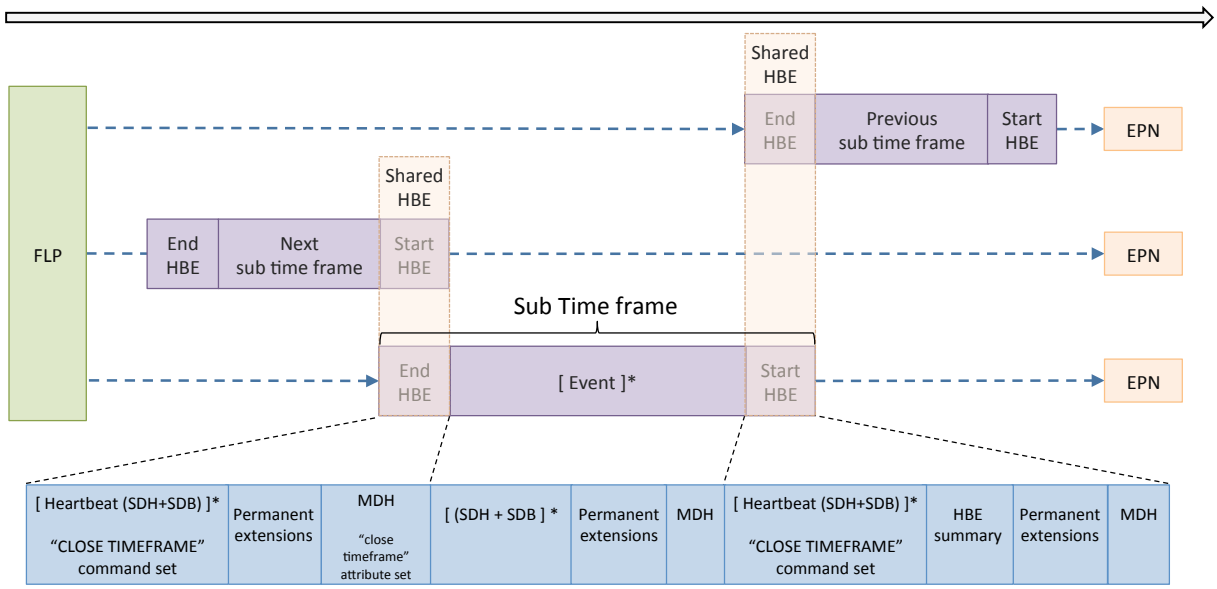

Figure 7.10: Format of the Time Frames between FLPs and EPNs.

be created locally on the FLPs mainly using data coming from the FEEs (usually two sets of triggered single block Heartbeats having the "CLOSE TIMEFRAME" command enabled).

Figure 7.10 shows an example where three time frames are transferred from one FLP to three EPNs. The component parts of the middle Time Frame are shown in detail.

"Start HBE" also contains a summary HBE specific header extension providing a detailed summary of all the data collected by the FLP during the associated Time Frame. On the FLPs, the HBEs contain both transient and persistent data: the transient portion is dropped when the data leaves the FLP while the persistent payload is kept throughout the whole lifetime of the event.

The HBE that closes a Time Frame sets a dedicated attribute (e.g. "END OF TIMEFRAME"), used by the EPN to identify the Time Frame boundaries. Usually the HBE events have to be duplicated in the output stream of the FLPs towards the EPNs, as the HBE that will close one Time Frame will also used as "Start HBE" for the following Time Frame. Offline handling must therefore be prepared to have in the collected data multiple HBEs tagged by the same timestamp. In the anomalous case of a FLP receiving a data block after that the corresponding Time Frame has been already closed and sent to the appropriate EPN, the FLP will include the data in the current Time Frame raising an appropriate error flag in the MDH like "OUT OF TIMEFRAME". The EPN must then be ready to handle such a condition according to a predefined policy (e.g. drop-and-report).

7.5.5 EPN data format

Each FLP has to send any STF tagged by the same HB to a common EPN. For fast navigation between the Time Frames on the EPN, it is foreseen to have folder-like navigation support. The Time Frame descriptor summarises the information in each sub-Time Frame by making it look like a directory. It will contain at the least the following fields: the FLP identifier; time information (start, end HB); links summary; IDs; error status; number of MDBs of different types; pattern of fired triggers. The status and error conditions are propagated in the processing chain embedded in MDH headers to avoid messaging bottlenecks and to associate them permanently to the data objects. The Time Frame has to be instrumented with search procedures based on indexing by MDH type, time, link ID while allowing for fast and vectorisable reconstruction.

At the end of the synchronous phase, the data are stored in CTF format. The asynchronous processing

produces ESDs and AODs of invidivual physics events.

7.6 Data quality control and assessment

In line with the general QC architecture description (Chap. 5.5) a design is presented (see Fig. 7.11) that will meet the requirements of the detectors and users. It ensures an efficient decoupling between the different QC processes. A slow or misbehaving task does not directly affect the other QC tasks as they have no connections or bounds between them. The key elements are described in the subsections below.

7.6.1 Data collection and QC data production

The architecture presented in Fig. 7.11 differs slightly from the schema shown in Fig. 5.7 as the "Generation of a monitoring object" will be done within the data flow itself as well as in an independent step. As shown in Fig. 7.11 the QC objects are produced in 3 different places:

1. Within other processes from the data taking pipeline like "Calibration 0" on the top left of the figure. This case applies in priority to the tasks which would anyway produce QC objects during their execution. It is also used when monitoring the decoding, calibration or reconstruction processes themselves. It has the advantage of saving resources by not duplicating the work but it could affect data taking and so should be used with care.

2. As independent processes on the EPN/FLP. Within the synchronous data flow (e.g. see "RAW QC"), these processes can have shared memory access to the data going through the node and will process as much as they can while the data are still accessible in the buffers but without slowing down the data taking. They run on key nodes and must behave properly within the allocated resources. When run asynchronously, see "CTF QC" in Fig. 7.11, the execution time is no longer a critical constraint and QC can potentially process 100% of the data.

3. As independent processes in dedicated QC nodes, see "Advanced QC" in the figure, these processes have less constraints because they cannot affect the data taking. A side effect is that they receive less samples, possibly around 1% of the full data. This approach is favoured for QC tasks that consume many resources (e.g. memory or CPU).

The system is flexible enough to manage QC object producers with different needs in terms of resources, performance, stability and data sample quantity. It will also accept objects produced in non-QC processes.

7.6.2 Merging

Most of the QC objects which are produced on the FLP and EPN nodes need to be merged.

As far as the FLPs are concerned, as QC is often by detector, merging could happen for a subset of the FLPs which serve a given detector but not all of them. The merging procedure can be done either by embedding QC results in the data stream as separate blocks, or using a scalable merging procedure as will be done for the QC objects of the EPNs.

On the EPNs, the QC objects will be produced while processing aggregated Time Frames. Since successive Time Frames will be reconstructed by different EPN nodes, the merging process will have to collect the QC objects from all EPNs. This merging can be executed synchronously for the ratio needed for fast quality assessment and asynchronously for the rest of the data. It means that the QC objects can still be populated and produced even though the mergers might not be able to cope with the amount of data.

Given the high number of inputs (up to 1 per EPN, i.e. 1500) a multi-level approach, such as Map-Reduce, will be used. Mergers and producers will be decoupled and each will be unaware of the existence of the

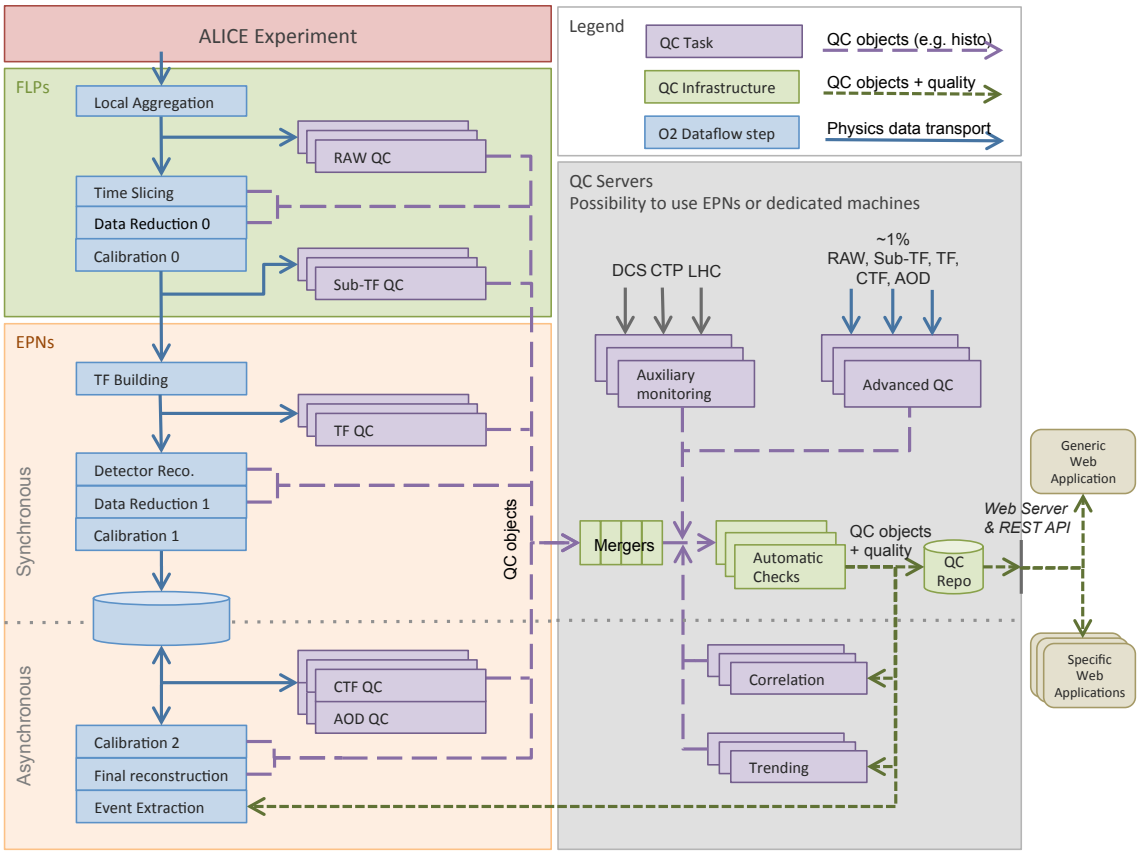

Figure 7.11: Quality control and assessment general design.

other. Producers will populate objects and make them available whenever possible. Mergers will pull the latest version of the QC objects at regular intervals and merge them.

7.6.3 Automatic checks

Automatic checks usually consist in the comparison of the QC data with a reference value, a threshold or a distribution. In general, references may change according to the run and data taking conditions including: detector hardware status; beam conditions or the collision system. For this reason, the QC is able to set and modify the reference online. For the sake of the reproducibility of quality assessment and for asynchronous use, the reference is stored and versioned in a database.

There is a dual interaction with the CCDB: the system allows for the reference data set online to be written in the database; the previously set reference and thresholds but also the variables related to data taking conditions are read from the database by the QC processes. Both synchronous and asynchronous QC processes need to access the reference in the database.

Separating the generation of QC objects from the quality assessment means that the latter can be done asynchronously and even iteratively if the reference data or the evaluation procedure were to be modified.

7.6.4 Correlation and trending

Some QC processes need to take as input the merged, and often time-aggregated, output of other QC processes. This is true for correlation and trending which can be considered as standard QC processes running within the QC farm. Despite the fact that they run asynchronously and should not need any merging, automatic check procedures need to be done and quality rated.

7.6.5 Storage

Merged QC objects and their associated quality are stored for future reference and in view of correlation and trending (see Sec. 7.6.4). The storage is able to handle the foreseen load which consists of an average of 5000 objects, peaking to 10000, updated every 60 seconds while serving the clients. A relational database is typically capable of such performance. Moreover, the limiting factor for such a system has been demonstrated to be the bandwidth [6]. For a pessimistic average size of 1 MB per monitoring object, a 10 Gb/s input connection would be far enough even for the peaks. Finally, such a system is able to cache seamlessly the data in case of congestions.

7.6.6 QC Results

The QC objects and their associated quality are used to check the transition from compressed to fully compressed data in the asynchronous part of the data flow.

During data taking operations, the QC results are accessed by shifters and specialists, inside and outside the control room, via one or several dedicated web applications. The system provides a web API, such as a RESTful API [7], to allow individual access to the necessary information for tool building.

A generic application is provided which meets the needs of the shifter and the common needs of the users in general. This application is also web-based and allows the display and manipulation of objects in a dynamic way. This implies the ability to transfer and display ROOT-based objects in a web browser. It uses the recent developments in ROOT which allow the serialisation of ROOT objects and files either in binary or in a JSON format. Most of the application is done client-side in javascript.

As the visualisation clients connect via a web server to the database, they cannot affect other QC processes. The web server can even limit the number of requests going to the database which could indirectly impact the QC chain.

7.6.7 Flexibility

The proposed design provides a maximum of flexibility for running on different nodes in the O^2 farm. For instance, a QC agent that can process Sub-Time Frames on the FLP nodes should also be able to run transparently on EPN nodes. This allows the dynamic balancing of the workload and postpones the execution of QC tasks in the data flow in the event that the FLPs are busy aggregating data.

The monitoring of CTFs can be delayed until resources are freed, even days after the end of the run. In this case, a percentage of CTFs would still be monitored immediately to give feedback to the shifter. Therefore, the system can do an immediate partial QC, with the partial calibration available at this stage, completing it later when more resources and more accurate calibration data are available.

7.6.8 Event display

A new event display will be designed for the Run3 in order to ensure a good use of the O^2 software framework. The event display is described in this report because it will interface to the new framework and because it is an essential too for the quality control and the commissioning.

The architecture of the Event Display System for Runs 3 and 4 has been dictated by the following main design requirements:

- Stability;
- Single Event Display switching between data sources;
- Features like bookmarks or event history browsing.

The design of the Event Display System is shown in Fig. 7.12. It introduces the Event Display Data Manager, which is the central part of whole system. It fetches and then stores data temporarily (Temp) adding entries to the database (DB). If the data are bookmarked, they are moved to permanent storage (Perm). Simultaneously, the Event Server communicates with client applications, providing information stored in the database and sending the requested data back.

This pattern helps to increase stability by decoupling the data sources and the clients. It also makes it possible to browse event history and makes it easy to receive, store and provide events requested by clients from different sources and with different formats. Depending on the future implementation of the main O^2 data storage, a link rather than the CTF itself could be stored. For other data types (e.g. TF or AOD), it would remain necessary to store them temporarily.

This architecture will be tested during Run 2. In parallel, another project called Total Event Display (TEV) will be developed. TEV is managed by CERN MediaLab and will provide a multi-platform solution for visualisation of events for any experiment. It will be tested and adapted to needs of ALICE during Run 2. After comparison, the candidate system more suited to ALICE will be installed for Run 3.

The interface between data repository and client applications will be especially important in case of use of TEV, which requires data in XML instead of ROOT files. The Event Display should not access data directly. In such a scenario, the Storage Manager could be partially adapted to act as the interface between the data repository and client applications. A final decision on this matter will be made at a later stage.

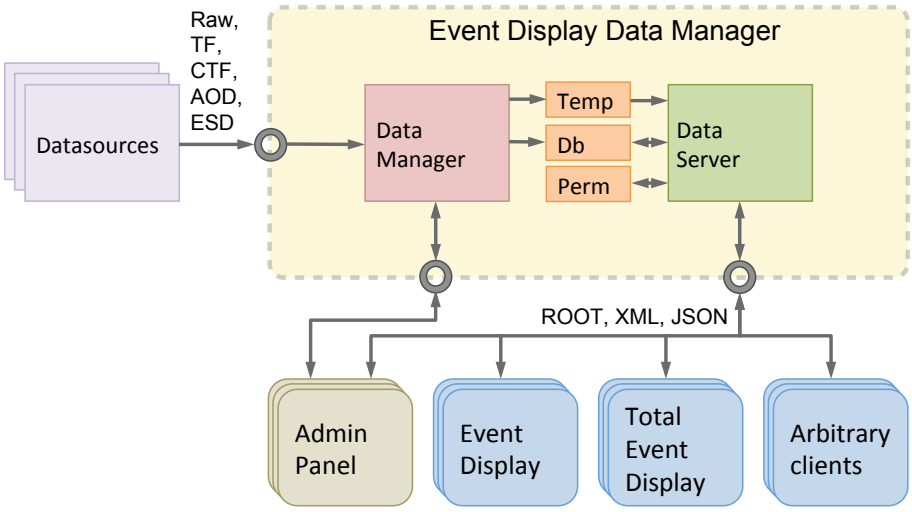

Figure 7.12: Event Display design.

7.7 DCS-O^2 communication interfaces

f

The data exchanged between the DCS and the O^2 system can be divided into two categories: the conditions and the configuration data.

The configuration data are sent to devices at different stages. Firstly, static configuration is loaded to the devices at startup of data taking, defining, for example, the assignment of the channels to detector modules or the composition of groups that will be operated together. Following this stage, the dynamic configuration carries the channel settings for each instance, trip or monitoring limits or alert thresholds.

These data are reloaded each time the detector configuration changes, as during Ramp Up. All configuration data are retrieved from the DCS configuration database. A subset of these, like noise maps or channel thresholds is required by the O^2 system and is provided along with the conditions data.

The conditions data are collected from devices such as temperature or humidity probes, power supplies or frontend cards. Detector conditions data required by O^2 represent about 10 % of the parameters supervised by the DCS. To isolate the O^2 system from DCS implementation details, a Data Collector process is implemented. The Data Collector connects to all detector systems and acquires available conditions data. The data exchange mechanism between the collector and detector systems is based on a publisher/consumer paradigm. Any change in monitored values is pushed to the Data Collector by the data publisher.

Collected values are stored in a formatted memory block, called the Conditions Image (CI). A dedicated O^2 process retrieves conditions data from the CI and inserts them into the DCS data frames to be injected to the O^2 system via a dedicated FLP. For each conditions parameter the alias, the measured value and measurement timestamps are stored.

The Data Collector maintains a list of parameters to be provided to the O^2 system. At startup, the Data Collector consults the DCS configuration and Archival databases and finds the physical location of each datapoint. With this information, it establishes connections to individual systems and subscribes to published values.

The frontend modules connected to the O^2 system via the GBT links are in a special category of devices. Physical access to these devices is achieved via the FLPs, which are not controlled by the DCS. A dedicated interface based on client-server architecture is implemented both on the DCS and FLP sides. DCS data produced by the frontend modules are transmitted to the CRU in dedicated DCS frames which are interleaved with standard data traffic. An FLP side process strips the DCS information from the data stream and publishes the received values. The client process on the DCS side subscribes to the publications and injects all values into the standard DCS processing stream. The same mechanism is implemented for data which need to be sent from DCS to the frontend modules, including register settings and on/off commands. The DCS server contacts the FLP maintaining the physical connection with the target devices and sends a command and the required parameters to the listening client. The FLP side client ensures the transfer of this data to the target device over the GBT link.

The read-out of DCS data is not centrally triggered. Each controlled device provides its own mechanism, typically based on the pooling of individual channels, to obtain the values. The internal read-out frequency largely varies between the devices provided by different manufacturers. To save bandwidth, only values that have changed during the actual read-out cycle are published. Updating this data in WINCC OA is therefore different for each channel. On each value change, the Data Collector is notified and the Conditions Image is updated. For stable channels, the value update occurs in average once every few seconds.

To create the initial Conditions Image, the Data Collector contacts all relevant WINCC OA systems and retrieves all current values. The value update request is executed for each conditions parameter at least once, at the Data Collector startup. During operation, the collector can access the DCS at several points as shown in Figure 7.13. If high update frequencies are required, the access points can be implemented outside WINCC OA, for example, at the level of the OPC server. After any change, each access point pushes data to the Data Collector.

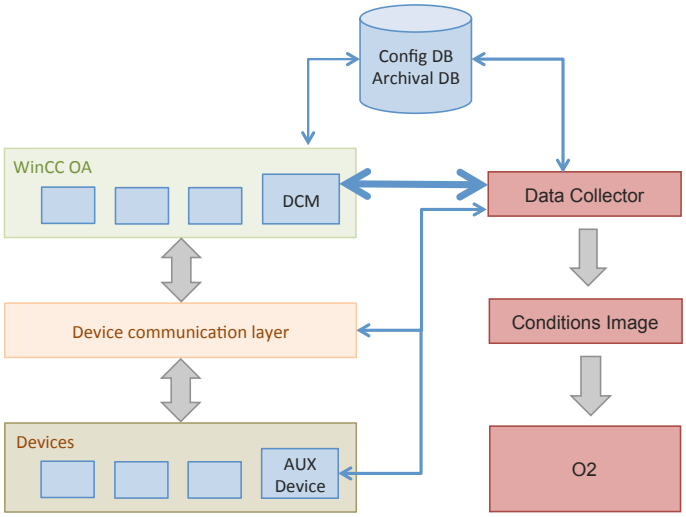

Figure 7.13: The DCS Data Collector and DCS Access Points.

A dedicated Data Collector Manager (DCM), implemented in WINCC OA, serves as a main access point for the Data Collector and covers most of the O^2 needs. It receives all value changes from WINCC OA and pushes them to the Data Collector. Laboratory tests have proved that DIM could be used as transfer protocol between the DCM and Data Collector with sufficient performance margin. The additional load introduced on WINCC OA systems by the DCM stays within reasonable limits.

For fast changing parameters, the device access layer can be used as a complementary access point and the Data collector can retrieve the values directly from the device drivers. This approach makes it possible to bypass the data processing in WINCC OA and provides instant access to the measured values.

Each stage of DCS data processing adds latencies. The biggest contributors to the delay between the physical value change and the published timestamp are the controls devices and the channel polling mechanism implemented in their firmware. For most parameters the effect is negligible, except perhaps for the fast detection of glitches. To overcome this limitation, dedicated measuring devices based on fast hardware have been installed. These devices monitor the fast changing parameters and publish them to the Data Collector. In parallel, all values are timestamped and sent to WINCC OA to be archived along with standard data.

Finally, an access point attached to the archival database gives access to historical values. The Data Collector can retrieve DCS data for any period of time and make them available to consumers. This working mode is reserved mainly for interfaces to external systems, such as the LHC.

Chapter 8

Physics software design

8.1 Calibration & reconstruction

In the following subsections, the steps of the calibration and reconstruction flow introduced in Fig. 5.6 are described in detail (see Sec. 8.1.1). The calibration and reconstruction procedures for different detectors are discussed in Sec. 8.1.2 and Sec. 8.1.3. Global tracking and event building are presented in Sec. 8.1.4 and Sec. 8.1.5. Finally, Sec. 8.1.6 deals with the requirements in terms of CPU and memory of the calibration and data taking procedures.

8.1.1 Calibration and reconstruction steps

In Runs 3 and 4, the calibration and reconstruction procedures will guarantee a maximum reduction of permanently stored data. As described in Sec. 5.4.1, the flow of these procedures can be split into five phases, which are schematically depicted in Fig. 8.1 (see also Fig. 5.6). These steps are discussed below in more details than in Sec. 5.4.1.

- **Step 0: processing on FLPs.** FLPs carry out low-level standalone processing (clusterisation, masking, calibration) of data supplied by the parts of detector they serve. The output of this step is Sub-Time Frames that are shipped to the EPNs and contain partially compressed clusterised or raw data. Additionally, since the calibration is affected by a detector's running conditions, a dedicated FLP will collect data from the DCS, processing and feeding them to the EPNs together with detector data.

- **Step 1: detectors' standalone processing on EPNs.** This is the stage where the final data reduction will be achieved. Standalone track-finding is carried out for the detectors concerned (ITS and TPC) both for data reduction and calibration purposes.

  At the very beginning of a fill only the calibrations from the previous fill will be available. As a consequence, in case the standalone processing needs calibration from FLPs, either some delay will be introduced between FLPs and EPNs in order to prepare such calibrations, or the first data shipped to the EPNs will not be processed with the most recent calibrations. In the latter case it is considered to be possible to reprocess the data concerned. The output of this step is filtered detector data sent to the permanent storage in the form of CTFs. Since the compression algorithms rely on the reconstructed track, they are also stored permanently in dedicated containers (ESD). At this stage calibration data will be buffered on a dedicated CCDB server and will be ready for aggregation at the end of the synchronous processing period.

- **Step 2: tracking.** A first ITS-TPC matching is performed at this stage, as well as the TRD tracking using TPC tracks as seeds. In addition, a sample of high p_{T} tracks will be extracted from the ITS-

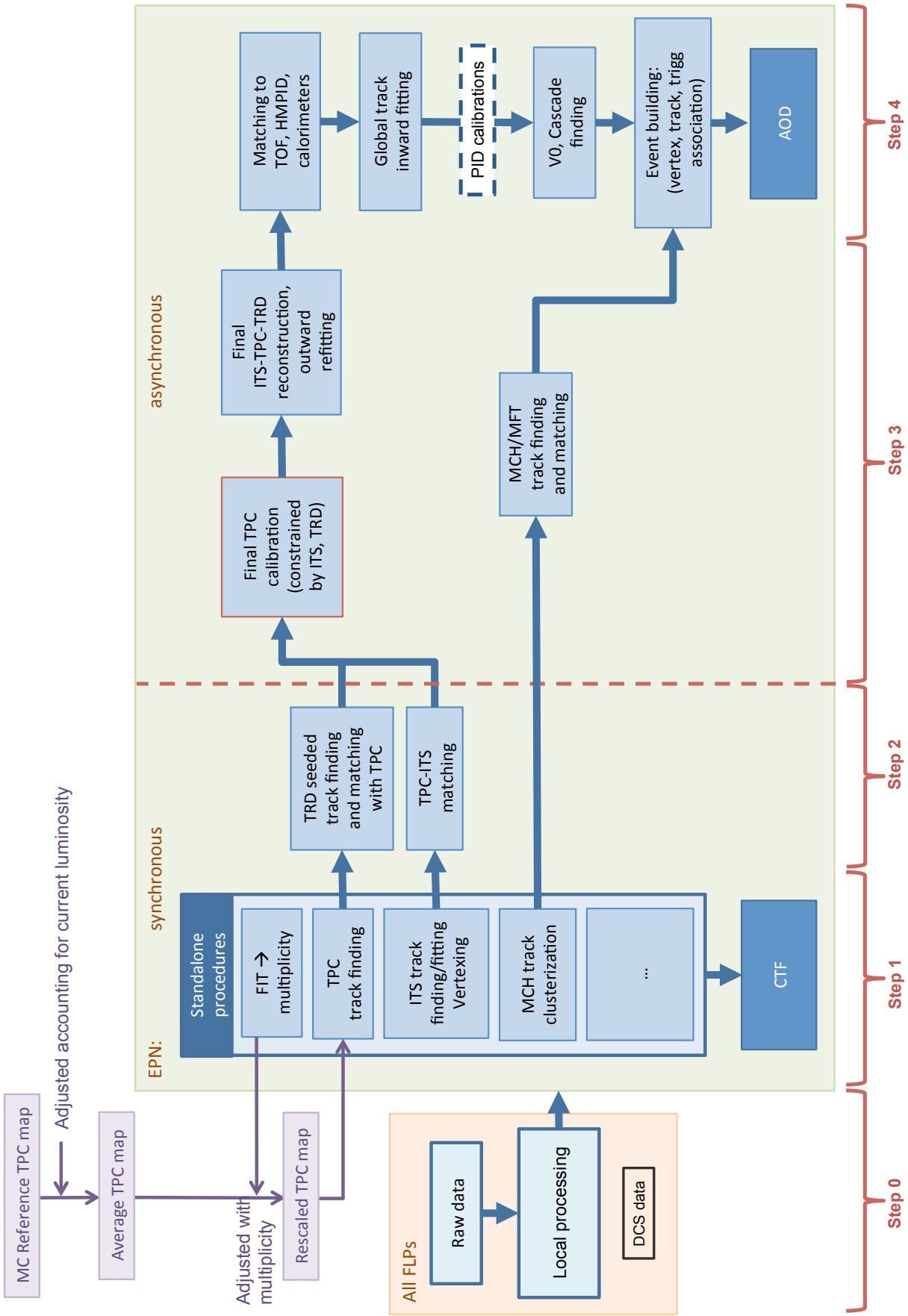

Figure 8.1: Schematic outline of the reconstruction and calibration data flow.

TPC-TRD matching to provide the rescaling factor for the Space-Charge Monte Carlo reference map to be adjusted to the current data taking conditions. Despite the fact that the full ITS and TRD tracking will be done in the later asynchronous processing, such a low frequency sample can be reconstructed synchronously at a low cost.

– **Step 3: final calibrations.** Here the TPC and TRD will be calibrated to the final quality level. The last ITS-TPC-TRD matching takes place at this step, together with a refit of the tracks enhanced by the updated calibrations. This step will also accommodate the track finding in the MFT and MCH detectors and their matching to each other. It will write the calibration data and feed track data for the final step.

– **Step 4: event extraction and AOD productiuon.** The last step of the data flow includes matching with the outer detectors, calibrations that require the best barrel tracking performance and in particular those to provide PID information, FIT reconstruction, and, finally, event extraction. The output of step 4 will be the AOD.

8.1.2 Calibration procedures

In the following subsections, some details about the calibration strategies of the ALICE detectors are given, focussing on the most complex ones. Tables 8.1 to 8.7 summarise the calibrations needed for different detectors and the type of processing they need (e.g. FLPs, EPNs). Calibrations done on the FLPs will occur during the synchronous phase of the process. Calibrations on the EPNs can be done synchronously or asynchronously. The mode will be indicated unless the calibrations are carried out in dedicated standalone runs or they require accumulated statistics.

TPC

The most demanding detector in terms of calibration is the TPC. As discussed in [1], the calibration requirements for this detector vary according to the expected performance. Two main stages can be identified, as outlined later in this section.

The most important adverse effect on this detector is the space charge distortion which can reach a peak value of $\sim$20 cm in r direction (and 7 cm in $r\phi$) [1]. The final TPC calibration will reduce this effect to the intrinsic detector resolution of a few hundred μm.

Data volume reduction is carried out by synchronous processing on the EPNs. At this first stage, the quality of the calibration is considered to be sufficient to perform a cluster to track association precise enough to obtain cluster corrections that allow the reduction of space charge distortions to the level of the intrinsic cluster resolution, i.e. O(1 mm). To achieve this result, the main requirement is a long term (O(15 min)) average map of space charge distortions, adjusted to take into account the current data taking properties such as luminosity and detector configuration ("Average TPC map" in Fig. 8.1). Two possible scenarios are foreseen for this processing step:

1. A subset of one or more EPNs is dedicated to the ITS-TPC-TRD matching to prepare the average map from the reference one to be then propagated back (possibly through a CCDB server) to all EPNs (including themselves) to perform standalone reconstruction for data reduction. This would have the advantage of being a single object used on all EPNs. Using a limited number of EPNs will reduce the time necessary to prepare the calibration;

2. Each EPN collects the ITS-TPC-TRD matching sample to produce the average map when processing the data that the EPN itself will receive. This approach would have the advantage of decreasing the amount of data to be exchanged, with the drawback of having to account for various objects from different Time Frames that reach different EPNs in a non-predetermined order.

Table 8.1: Summary of the different detector calibrations for TPC and ITS.

Detector	Calibration	Processing type	Beam [y/n]	Freq.	N. of events
TPC	Pedestal/Noise	FLP	n	fill	O(100)
	Pad Gain	Standalone runs	n	year	-
	Drift Velocity	EPN (sync)	y	15 min[1]	O(1k)
	Space Charge	EPN (sync)	y	15 min[1]	O(5k)
	Drift Velocity	EPN, (async[2])	y	15 min	O(1k)
	Gain	EPN (async[3])	y	15 min	O(1k)
	Space Charge	EPN (async[4])	y	5 ms	all
ITS	Noisy/dead/faulty channels	Standalone runs	n	run	O(1k)

A more granular rescaling for the event and track multiplicity information from the previous $\sim$160 ms (corresponding to the maximum ion drift time in the TPC) will also have to be applied ("Rescaled TPC map" in Fig. 8.1). For this purpose the information about ion current in the TPC will be collected and kept on the FLP before being transferred to the EPNs and embedded in the data. Drift velocity for TPC is calibrated every 15 minutes.

Other calibrations independent of time will also be considered at this stage. Typically these reflect detector conditions like pedestal values and dead channels maps. The pad-by-pad gain equalisation coming from dedicated calibration runs will also be needed.

After the first step described above, the TPC data will be used in a synchronous standalone tracking procedure ("TPC track finding" in Fig. 8.1, and see 8.1.3) aimed at reaching the necessary data volume reduction factor needed for efficient data storage. The following step can be performed asynchronously, allowing for more CPU and time consuming tasks, the aim of which is to provide the full tracking resolution for physics analysis with a space charge correction at the level of the TPC intrinsic track resolution of 200 μm.

In order to reach this level of precision, further space charge correction maps will need to be calculated with high granularity in space and time. The update interval for these maps is estimated to be on the level of a few ms (O(2-5)) and driven by the remaining local fluctuations ($\sim$1%) of the space charge which are not already included in the average rescaled map used for the first part of the processing [1]. To follow the fluctuations it is foreseen to have information on the read-out currents of the TPC during the last 160 ms integrated in steps of 1 ms available during the reconstruction (with a different granularity than their usage for the average map).

Additionally, the interpolation of track segments from the ITS and TRD will be used to calibrate residual distortions ("Final TPC calibration (constrained by ITS, TRD)" in Fig. 8.1). This will be possible since the level of calibration available for these detectors at this stage will be sufficient for such a procedure. The necessary data for this task will be collected during the synchronous processing. The calibrations for the drift velocity and the gain will also be taken into account with greater precision than in the previous standalone tracking step. For the gain, the calibration data will be prepared during the synchronous step, as for the high-granularity space-charge distortion correction map.

[1]The update could occur more often depending on the stability of the data taking conditions (e.g. luminosity).

[2]This calibration assumes that the synchronous drift velocity calibration will be merged, and made available to all EPNs. A further improvement of this calibration will be obtained during the calibration of space charge through a fitting procedure.

[3]Data for this calibration will be collected during the synchronous phase to be used then for the asynchronous one where a scaling as a function of the high voltage settings, and pressure and temperature (P/T) will be used. It is assumed to have stable

Table 8.2: Summary of the different detector calibrations for TRD.

Detector	Calibration	Processing type	Beam [y/n]	Freq.	N. of events
TRD	Pedestal/Noise	FLP	n	fill	100
	t_0	FLP (monitoring on EPNs)	y	run	O(10k)
	Chamber Status	EPN	y	run	O(35k)
	Drift Velocity	EPN	y	run	O(3.5k)
	ExB	EPN	y	run	O(3.5k)
	Chamber Gain	EPN	y	run	O(35k)
	Pad by pad gain	standalone runs	n	year	O(2G)
	PID reference	Asynchronous	n	period	all

ITS

ITS calibrations concern the identification of noisy/dead pixels and will be performed on the FLPs during standalone runs, without imposing any constraints on the TPC calibration procedure. Moreover, the identification of the faulty pixels that demonstrate problematic behaviour would be carried out at the same time.

TRD

The main parameter to be calibrated for TRD in order to perform track finding is the time reference (or offset, t_0) with respect to which the r-coordinate of the TRD cluster is obtained ("TRD seeded track finding and matching with TPC" in Fig. 8.1). The time offset depends on the triggering conditions in the experiment and on the TRD FEE configuration, and enters the tracking as a shift of the points in the radial direction. The t_0 calibration will be performed on the FLP using the measurement of the pulse shape (height) of the signal from the raw data acquired in usual physics runs. Calibration on this level will allow the TRD to contribute to the final calibration of the space charge distortion (and drift velocity) in the TPC.

The remaining TRD calibrations will include the status of the TRD chambers, the drift velocity, the Lorentz angle (ExB) and the gas gain. The identification of malfunctioning chambers to be excluded from the tracking will be carried out on the EPNs. An algorithm that checks the detector occupancy will be used together with the information on the drift velocity, ExB and gas gain calibration. The drift velocity and ExB will be derived from the matching with the TPC and can rely on the TPC performance after the first level calibration. The gain calibration, which is crucial for particle identification, will be done at different levels. The gain calibration on the chamber level will be carried out at the same time as the drift velocity and ExB calibrations on the EPNs. The pad-by-pad gain calibration will need a dedicated run with Krypton with high statistics in order to reach the required precision. It will therefore be done only once per year. Finally, the particle identification references will be extracted from a larger amount of data, usually this will cover a whole period.

The TRD will also take short standalone runs to measure the noise and calibrate the pedestals. These will be processed on the FLPs.

Figure 8.2 summarises schematically the calibration and reconstruction inter-dependencies between ITS, TPC and TRD, as just described.

ion backflow conditions in the detector. If this is not true, a higher granularity with respect to the one in the table will be needed (update frequency of 20s using 1M events).

[4]Data for this calibration will be collected during the synchronous phase to be used then for the asynchronous processing.

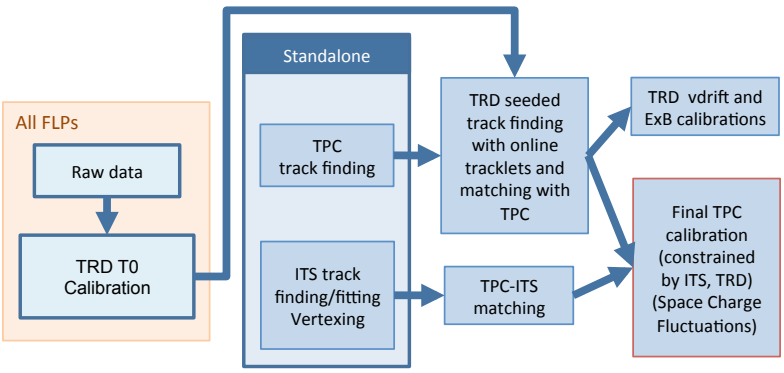

Figure 8.2: Schematic outline of the reconstruction and calibration data flow.

Table 8.3: Summary of the different detector calibrations for TOF.

Detector	Calibration	Processing type	Beam [y/n]	Freq.	N. of events
TOF	Channel Efficiency	FLP	y	run	all
	Detector Status	EPN	y	run	[5]
	Noisy Channels	Standalone runs	n	fill	O(100k)
	Time Calibration	Accumulation on EPNs and Commissioning run	y	year/period	O(1M)/O(10M)[6]
	"Problematic"	Accumulation on EPNs and Commissioning run	y	period	O(1k)

TOF

The TOF calibration will update, when necessary, the TOF channels status map as received by the declared configured electronics. Additional channels, identified as noisy in dedicated standalone runs at the beginning of each fill, or found to be not functioning properly during data taking, might be tagged to be excluded later by the physics analysis. These procedures will be running on the FLPs accumulating data, and exporting the final map at the end of the run. In addition, the calibration will include the measurement of single channel time offsets and time-slewing corrections and the identification of the channels with an unstable time response. These steps will be run on the EPNs. Due to the high statistics required for channel offsets and time-slewing calibrations and for identifying problematic channels, it will be necessary to accumulate the information during data taking followed by further processing. This means that TOF calibrations will only be used at analysis stage. As a consequence, this will limit the TOF reconstruction by postponing the matching stage until the final calibrations can be applied.

MCH and MID

The only calibrations foreseen for the Muon spectrometer are to determine occupancy (for the MCH) and noisy channels (for the MID) maps which will both be calculated on the FLPs.

[5] This calibration will be obtained from DCS data so no dependence on the number of events is given.

[6] Two numbers are quoted here in order to account for two different calibrations. The one requiring the highest statistics will be done with a much smaller frequency (see "Frequency" column).

Table 8.4: Summary of the different detector calibrations for MCH, MID, and MFT.

Detector	Calibration	Processing type	Beam [y/n]	Freq.	Number of events
MCH	Occupancy	FLP	y	run [7]	O(10k)
MFT	Noisy/dead/faulty channels	Standalone runs	n	run	O(k)
MID	Noise	FLP	y	run	O(1k)

MFT

Like the ITS, the MFT will create maps for noisy/dead/faulty channels on the FLPs during dedicated standalone runs.

FIT

The FIT detector will need three types of calibration: slewing corrections (with laser data or data collected at the beginning of the data taking); channel equalisation (on the FLPs, using the first O(1000) events, or using commissioning data); global offsets (to align the measured times to zero). The latest calibration is time-dependent, and will also be performed on the FLPs. Moreover, the measurement of the event multiplicity will need the calibration for the number of charge particles as a function of signal amplitude. The starting point for this calibration will be the output of standalone laser runs. This will be used for the subsequent synchronous fitting procedure on collision data, which will be possibly improved in the asynchronous stage.

Table 8.5: Summary of the different detector calibrations for FIT.

Detector	Calibration	Processing type	Beam [y/n]	Freq.	Number of events
FIT	Global Offset	FLP [8]	y	run	O(1k) [9]
	Multiplicity Calibration	Standalone runs FLP [11]	n	d [10]	
	Time Slewing	Commissioning run	y	2-3/yr	
	Channel E	Commissioning run	y	month	O(1k)

HMP

For every run, HMP will perform calibrations on the EPNs for chamber gain and the refractive index, based solely on the input provided by DCS. The pedestals will be calculated in very short (2-3 minutes) dedicated calibration runs before every fill and in physics runs, their subtraction will be performed on the read-out electronics.

[7] The frequency of this calibration could increase and become more frequent during a run.
[8] Possible only if FIT is read out by one FLP only, otherwise, EPN needed.
[9] The events mentioned for this calibration are FIT triggered events.
[10] The frequency will decrease once the procedure is well established.
[11] Possible only if FIT is read out by one FLP only, otherwise, EPN needed.

Table 8.6: Summary of the different detector calibrations for HMP and ZDC.

Detector	Calibration	Processing type	Beam [y/n]	Freq.	Number of events
HMP	Chamber Gain	EPN (async)	y	run	[12]
	Refractive Index	EPN (async)	y	run	[12]
	Pedestal	Standalone runs	n	fill	O(1k)
ZDC	Energy calibration	Standalone runs	y	2-3 days	O(1k)

Table 8.7: Summary of the different detector calibrations for Calorimeters (EMC, PHS) and CPV.

Detector	Calibration	Processing type	Beam [y/n]	Freq.	Number of events
Calorimeters (EMC, PHS)	Energy Calibration	Accumulation on EPNs or AODs	y	1-2 periods	O(200M) [13]
	Temperature dependence of gain [14]	Accumulation on EPNs	y	run	[15]
	Time Calibration	Accumulation on EPNs or AODs	y	period	O(10M) [16]
	Bad Channel Map	Accumulation on EPNs or AODs	y	period	O(1M)[16]
CPV	Pedestal calculation	FLP	n	fill	O(2k)
	Gain Calibration	A on EPNs	y	[17]	O(10M) (pp MB) [18]

ZDC

The energy calibration for the ZDC will be obtained in standalone runs where ZDC triggers will be used.

Calorimeters (EMC and PHS)

The energy and time calibration of the EMC detector will be based on the reconstruction of the π^0 peak performed in every cell and on the distribution of the times measured by each cell, respectively. Large statistics are required for such calibrations; (O(10^8) EMC L0 triggered events). These together with the need for several passes over the data (4 passes for energy and 2 for time calibration), makes them unsuitable for online calculation. As a consequence, an accumulation procedure on the EPNs followed by an offline (after data taking) analysis of the data will be necessary. The same approach will be followed for the energy and time calibrations of the PHS detector. The time dependence of the gain of the EMC towers, which can be considered as a second-order energy calibration, will be accounted for using DCS data. Finally, both EMC and PHS will produce a bad channel map for Monte Carlo simulations. This will be built offline, based on data collected on the EPNs. The strategy for the calorimeters' calibration will not impose any constraint on online processing as the EMC calibrations are needed only at the analysis level. It should be noted that the accumulation of data on the EPNs could be substituted by the direct use of the final AOD-like objects.

[12] This calibration will be obtained from DCS data so no dependence on the number of events is given.
[13] EMC needs EMC-triggered events for this calibration.
[14] EMC only. This calibration will be obtained from DCS data so no dependence on the number of events is given.
[15] The data for this calibration will come from DCS, so no number of events is given.
[16] EMC needs EMC-triggered events for this calibration.
[17] To be studied during Run 2.

CPV

As CPV has similar amplitude electronics to the HMPID, it will need a pedestal calibration run of about 2 minutes before each fill. The runs will be processed on the FLPs. During data taking, the pedestal table will be loaded to the front-end electronics for further zero suppression. Gain calibration will be calculated on the EPN using physics data by equalisation of the most probable value of the ionisation loss spectra per channel. Statistics needed for the gain calibration have to be large enough to cover a few hundred hits of charged particles per CPV channel. Since this corresponds to several days of data taking, the data will be accumulated on the EPN and once enough statistics is obtained, the calibration will be made ready for future use.

Special calibrations strategies

- Standalone calibrations: As in Run 1, some calibrations will take place in dedicated runs (STANDALONE runs). This is a requirement for the CCM and is addressed in 7.3. The output of such calibrations will be needed either before the beginning of the next PHYSICS run, for synchronous processing, or at a later stage, for asynchronous processing (e.g. analysis). Standalone calibrations can take place on both FLPs and EPNs, and particularly the latter, if full detector information is required.

- Calibrations requiring large statistics: if these cannot be obtained during a single run, synchronous processing should allow for the accumulation of the necessary data. A post-processing task is then be able to run over such data and produce the calibrations required. This approach will include only the calibrations that are not needed for any stage of the synchronous processing. An example is the energy and timing calibration for the calorimeters. Typically, such a requirement arises from the need to properly simulate Monte Carlo samples that take into account the detector efficiency averaged over a large time interval. Examples include bad channels and detector hardware response.

8.1.3 Reconstruction procedures

TPC

The zero-suppressed raw data (pad, arrival time and signal amplitude information) is converted on the FLPs to clusters by a one dimensional FPGA algorithm running in the CRU. It processes the pads sequentially, finding the maxima within neighbouring pads on the same pad row.

Reconstruction on the EPNs will assign a collision timestamp t_0 to each cluster, necessary for the determination of its z coordinate and applying position dependent calibration. In absence of the trigger signal, the following method, shown in Fig. 8.3, has given satisfactory test results [1]: (i) A very rough estimate of t_0 is obtained for each cluster assuming that it belongs to a track with about half of the maximum drift length (i.e. $|\eta| = 0.45$). This reduces the maximal distortions by a factor of $\sim$2, thereby improving seeding; (ii) Then, for any seed made of two clusters, the t_0 is adjusted in such a way that the seed points on the central Luminous Region (LR). This reduces the uncertainty in t_0 to the typical size of the LR divided by the drift speed, i.e. to $\sim$3 μs. At this point, the t_0 estimate can be matched to one of the interaction timestamps from the list of interactions recorded by the FIT detector within the possible maximum drift time of 100μs. After this, the space charge distortion correction with average map rescaled for luminosity can be applied to the TPC clusters.

The next step is the standalone track finding using clusters with residual mis-calibration, performed by a fast and easily parallelisable *cellular automaton* approach currently used in the HLT tracking. Once the

[18] In case the calibration obtained from pp data will not be stable enough, a further recalibration for PbPb data will be needed, requiring a factor 400 less statistics of Minimum Bias events.

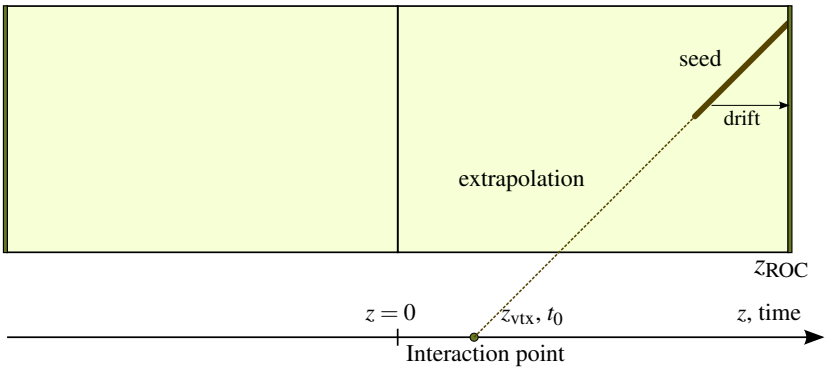

Figure 8.3: Schematic diagram of the seeding procedure and t_0 estimation.

track finding is carried out, an after-burner algorithm will search for the large curvature *looper* tracks, corresponding to secondaries of a momentum below ∼50 MeV/c. The clusters of these tracks do not contribute to physics analysis and constitute nearly 70% of total TPC occupancy. Their elimination greatly contributes to the reduction of the data volume. Once these synchronous reconstruction steps are carried out, the compressed cluster data are saved to permanent storage.

ITS

Cluster finding is done on the FLPs using a simple algorithm which searches for the adjacent pixels assuming certain row-column ordering in the shipped data.

Contrary to the current situation where most of the tracks in the ITS are found as a prolongation of TPC tracks, in Run 3 the TPC will need ITS tracks to carry out final calibration of the space charge distortions. For this reason, standalone tracking in the ITS is foreseen on the EPNs. The principal requirement for ITS tracking code is speed, since a partial reconstruction of high p_T tracks must be done synchronously to provide constraints for the TPC calibration. The algorithm that is currently under study is based on a Cellular Automaton (CA) [2][3].

TRD

As described in [4], TRD will be able to take data in different configurations. The default mode in Runs 3 and 4 will be the read-out of online tracklets, where the FLP stage is needed for their post-processing and for quality selection, like the marking of pad row crossings.

In the case of the full or partial raw ADC data read-out, the reconstruction will start on the FLP with the conversion of the pad and time bin raw data to clusters. Then these will be bound into tracklets (track segments within a single layer) which are connected to the full tracks on the EPN, using seeds from the TPC.

A third possibility consists in running the TRD in mixed mode, for example with online tracklets augmented by the raw data for a small fraction $\sim 10^{-3}$ of events. This will require the execution of both the operations outlined above.

In all cases, the reconstruction will be performed in synchronous mode in order to provide constraints for the final TPC calibration that will take place asynchronously. Reducing TRD data rates by discarding unusable online tracklets is being studied.

TOF

The reconstruction is performed on the EPN in asynchronous mode and consists of matching the TOF hits to the extrapolations of tracks from smaller radii.

MCH and MID

The reconstruction will start by pre-clustering on the FLPs. This means grouping adjacent fired pads, irrespective of their charges, currently taking up 20% of the total clustering time. It is carried out with an improved algorithm, 20 times faster than now. This is very interesting when coupled with the option of using the CRU FPGAs hosted by the FLPs. The next step is the synchronous clusterisation on the EPNs for data reduction. The possibility of performing full cluster finding on the FLPs has still to be investigated; the FLPs would need to see complete detection elements (either slats or quadrants). If this is the case, then converting the clustering algorithm to a GPU version would be worth considering which accelerates the process up by a factor of $\sim$50. The track finding will be performed at the asynchronous stage.

MFT

The reconstruction procedures for this detector are the same as for the ITS.

FIT

The synchronous reconstruction stage consists of:

- An estimation of the precise interaction time by averaging the mean arrival time on each side (first estimate using the first signals generated on the A and C sides is performed already in the FEE);
- The determination of the vertex position from the difference of the mean signal arrival times;
- Preliminary multiplicity estimates from the amplitudes of each Micro-Channel Plate (MCP) sector by applying corrections based on simulations.

ZDC

The reconstruction takes place on the EPN requiring the subtraction of the baselines and the summing of the amplitudes of different channels weighted with calibration parameters obtained in dedicated standalone runs.

HMP

Cluster finding from the fired pads, coordinates and charge will be performed on the FLPs and matched to barrel tracks at asynchronous stage.

Calorimeters (EMC and PHS)

Both detectors will extract the signal amplitude and the time in each cell on the FLPs by fitting the pulse shape, reducing an array of up to 30 10-bit samples to only two parameters: cell energy E and timestamp t. Both detectors will perform cluster finding in adjacent cells on the EPNs . EMC will apply temperature corrections using DCS data. The shower shape and cluster parameters, as well as cell signals and time information will be stored for the analysis stage.

CPV

The first step for this detector is the reconstruction of the pad amplitudes after pedestal subtraction and scaling by the gain calibration parameters. Then cluster finding will run on the EPN from the reconstructed pad amplitudes. As CPV is used for the neutral particle identification in PHS, cluster matching in CPV and PHS is performed, taking into account track matching with CPV clusters and their further propagation to the surface of PHS. Since particle identification is based on objects reconstructed in several detectors (CPV, PHS, ITS, TPC), the algorithm has to work at the EPN level.

8.1.4 Global tracking

After performing in asynchronous mode the final calibration of the TPC space charge distortions using constraints from the ITS and TRD ("Final TPC calibration (constrained by ITS, TRD)" in Fig. 8.1) the tracking in TPC, ITS and TRD is repeated to maximise efficiency; global tracks are refitted ("Final ITS-TPC-TRD reconstruction, outward refitting" in Fig. 8.1), and prolongated outwards to find the matching clusters in the TOF, HMP and calorimeters ("Matching to TOF, HMP, calorimeters" in Fig. 8.1). The barrel track-finding is completed by refitting the tracks inwards to the Luminous Region ("Global track inward refitting" in Fig. 8.1), and with the finding of the primary and secondary vertices. In parallel, the matching between the MCH/MID and MFT tracks is performed, followed by the propagation to the LR ("MCH/MID/MFT matching" in Fig. 8.1).

8.1.5 Event extraction

The final step of the asynchronous Time Frame reconstruction is the *event extraction*: attributing the tracks and secondary vertices to primary vertices and their association to trigger timestamps from the FIT. The ambiguity of such an operation in high pile-up rate conditions depends on whether a timestamp can be given to individual tracks. In the following estimates, a 50 kHz Pb–Pb interaction rate is assumed with the same beam structure as in [1]: 12 equally-spaced 2.34 μs bunch trains with 48 bunches each. This gives an interaction probability $\mu = 0.0079$ per bunch crossing and 0.38 per train crossing.

For the successfully matched tracks between the TPC and the ITS (more than 95% above p_{T} of $\sim$150 MeV) the time resolution is determined by the TPC track (p_{T} dependent) z uncertainty (at the outermost cluster of the ITS) divided by the drift speed. This resolution is $\sim$0.25 cm for the average track, making a time resolution of $\sim$0.1 μs, thereby reducing the pile-up (probability of having a track from a different collision in the same time interval) rate for such tracks from $\sim$50% in the ITS read-out window of up to $\sim$30 μs to $\sim$2% level. Given that with a typical $\sigma_z \sim$6 cm for the luminous region, less than 1% of pile-up collisions are separated by a distance smaller than 1 mm, and that the z-resolution after full calibration is better than 200 μm even for the global tracks of lowest momenta, finding unambiguous primary vertices and attributing them to one of the trigger timestamps from the FIT does not pose any problem. The same is true also for the association of global tracks (both primary and secondary, as well as the secondary vertices made of such tracks) to the corresponding primary vertex. Similar conclusions are valid for pp interactions at 200 kHz.

The situation is somewhat different for the ITS tracks which were not matched in the TPC (mostly tracks below 200 MeV), especially if the most pessimistic scenario for the ITS read-out cycle of up to 30 μs is assumed. In this case a single ITS read-out cycle will integrate on average 1.5 (6) collisions in Pb–Pb at 50 kHz (pp at 200 kHz). The only way to associate the ITS tracks from such piled up interactions with a specific trigger timestamp is via their relation to the primary vertex found with global tracks. The possibility of doing this depends on spacial vertex separation in piled-up collisions. Figure 8.4 shows for the randomly picked vertex the probability of having at least one pile-up vertex within a certain distance (Y axis) as a function of the number of piled-up collisions. Assigning a primary track to its vertex on the condition of its isolation from other vertices by at least 1 mm leads to ambiguous attribution in $\sim$1.5% collisions in Pb–Pb and $\sim$5.5% in pp. Increasing the vertex isolation condition to 2 mm (e.g. for the attribution of secondaries from heavy flavour decays) increases the fraction of ambiguous attributions by a factor of 2.

For the above reasons, the following event extraction strategy can be considered:

- The global tracks are sorted according to their matching with the FIT timestamps;
- The primary vertices are found with the requirement that only the global tracks with the same timestamp contribute to a given vertex; the same condition is set on secondary vertices made of

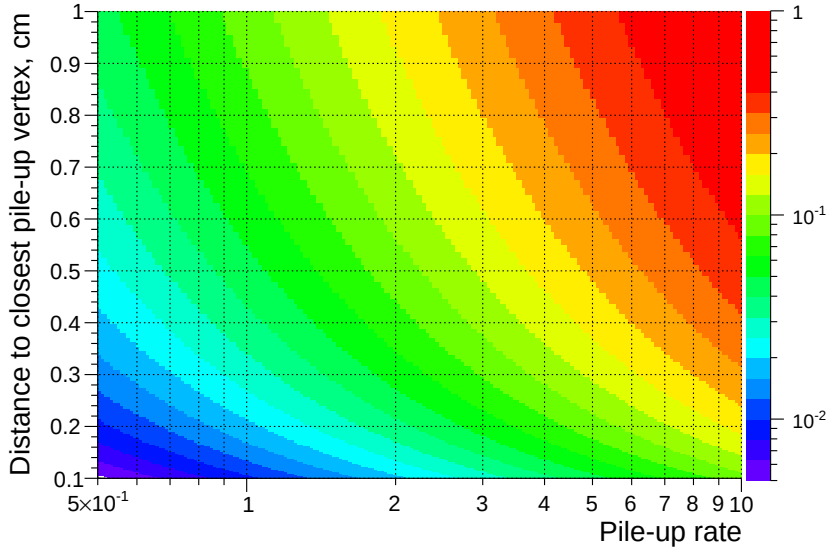

Figure 8.4: Probability of at least one pile-up collision happening in proximity (Y-axis) for selected vertex as a function of the pile-up rate (X-axis). Luminous region $\sigma_Z = 6cm$ is assumed.

global tracks;

– The secondary global tracks (not attached to any of the primary vertex at the vertexing stage) as well as the associated secondary vertices are attributed to one of the vertices using their time and distance to primary vertex information;

– The same operation is carried out for the unmatched ITS tracks and secondary vertices associated with them; in case of ambiguities in time and distance to vertex information, these tracks are tagged by the list of the FIT timestamps (or primary vertices identifiers) to which they can be attributed.

In this approach, the output of the event extraction will be the container of the primary vertices associated with the timestamps and the list of unambiguously attributed tracks as well as the separate list of tracks which can be associated with more than one vertex or timestamp.

The output of each reconstruction step should contain metadata identifying and giving a precise description of: (i) The software version used; (ii) The set of relevant calibration objects; (iii) The status of the reconstruction for each detector and its usefulness for physics analysis.

8.1.6 Computing requirements

Table 8.8 summarises the requirements of the most important calibration and reconstruction tasks. The CPU labels single CPU core.

8.2 Physics simulation

The objective of Physics Simulation is to provide large samples of events that resemble as closely as possible real data. The simulation process comprises the generation of the final states of pp, p–Pb and Pb–Pb collisions and the subsequent simulation of the passage of all generated particles through a matter distribution that represents as faithfully as possible the actual experimental setup. For those particles passing sensitive detector elements, the analogue detector response is simulated in specific user code and in a subsequent digitisation step converted into the raw simulated detector information used for real data (digits, raw data). These events are reconstructed like real data. In addition to the raw and reconstructed

Table 8.8: CPU requirements for processing the data from Pb–Pb interaction at 50 kHz in the synchronous mode.

Detector	Process	Processing requirement [CPU cores or GPUs.]	Processing Platform	System reference
TPC	Calibration	1000	CPU	Intel I7-4600U 2.70 GHz
TPC	Track seeding, following	5000	GPU	AMD S9000
TPC	Track merging, fitting	15000	CPU	Intel I7-980X 3.60 GHz
ITS	Tracking	75000	CPU	Intel I7-2720QM 2.20 GHz
MCH	Preclustering	200	CPU	Intel I7 2.30 GHz
MCH	Clustering	5000	CPU	Intel I7 2.20 GHz

data the full history of the event (event record, Monte Carlo truth) is stored. This makes it possible to evaluate the detector performance in terms of acceptance, efficiency and resolution from which correction procedures and their corresponding systematic uncertainties are derived. The computing time spent in simulation must be kept low enough so that, depending on resources, sufficiently large Monte Carlo data samples can be generated for a timely analysis of the data and subsequent publication of physics results.

8.2.1 Runs 3 and 4 simulation requirements

With an increase of the data volume by two orders of magnitude, the Runs 3 and 4 simulation requirements will notably increase. Taking into account priorities of the physics programme and the requirements of multiple analysis topics, it is estimated that the simulation requirements will increase by a factor of 20. Most of this increase will be met by using fast or parameterised Monte-Carlo simulations. For the remaining full simulations, it is essential to use a computationally efficient particle transport code without compromising for accuracy. This full simulation must be able to make efficient use of computing and opportunistic supercomputing resources available on the GRID/Cloud. At present, only the Geant4 transport package, with multi-threading enabled, can achieve this.

Optimised simulation strategy At the same time, a goal-oriented strategy has to be adopted to simulate only what is strictly needed. This strategy requires planning simulation in three stages: in the first stage, the simulation and reconstruction code is validated; in the second stage, enough events are produced to obtain acceptance, efficiency and resolution maps for single particles over a wide phase space volume; at the third stage, parameterised acceptance, efficiency and resolution are used for a fast parametrised simulation of multi-particle signals (N-particle phase space integration). In subsequent simulations the second stage can also be replaced by fast full simulations reducing the particle transport and detector response simulation to the minimum needed to perform a meaningful stage 3 simulation. Hybrid procedures combining full, fast full and parameterised simulations can also be envisaged.

Optimal usage of the resources needed for the second stage include trivial variance reduction and merging techniques. The merging techniques comprise:

- **Merged Kinematics** Embedding of signals in the background event during primary event generation
- **Merging with MC** Merging of generated signals using the summable digits of the background event
- **Embedding** Merging of a generated signal with real data

8.2.2 Transport code and the virtual Monte Carlo interface

Simulations performed for Run 1 used the transport codes Geant3, Geant4 and FLUKA interfaced to the Virtual Monte Carlo (VMC) [5]. The VMC interface insulates the user code from changes of the detector transport code. The replacement of the geometrical modeller of the different packages by a single modeller from the ROOT package represents a further level of generalisation and simplification for the user code. While Geant3 has been used for all our main physics production, Geant4 has been used for some specific applications and Fluka mainly for radiation studies.

Geant4 VMC MT Starting with Geant4 version 10: full support for multi-threaded simulation applications has been included. Parallelism is achieved at the event level, with events dispatched to different threads. Geant4 VMC represents the realisation of the VMC interface for Geant4. At the same time it can be seen as a Geant4 application implemented via the VMC interfaces. Starting from version 3 it supports Geant4 multi-threading. The MT mode is activated automatically when Geant4 VMC is built against Geant4 MT libraries.

Geant4 VMC MT and the scaling behaviour of the computing time with the number of cores have been tested using a simplified but realistic multi-threaded simulation application. It is based on Geant4 VMC standard example A01. Particles from parameterised Pb–Pb generation are transported through the full ALICE geometry including a realistic field map; hits are generated and stored for the TPC.

8.2.3 Detector response and digitisation

Further developments of the detector response and digitisation code will be driven by the requirements from the continuous read-out scheme - simultaneous snapshots or sequential ”rolling shutter” read-out. The main challenge here is to avoid breaking the trivial event parallelism. Sequentiality is introduced by the conversion of digits into the new raw data format (Time Frames) and by the need to simulate the detector response based on the event history. The most stringent requirement comes from the realistic TPC continuous read-out simulation: at an interaction rate of 50 kHz ions from 7500 events contribute to the space charge distortions and event-by-event fluctuations requiring a calibration every 2 ms. Even assuming that these specific simulations will be limited to relatively small event samples (3M Pb–Pb events every 3 months) a procedure must be implemented that allows for a maximum of parallelism in the production of these events.

Currently under investigation is a procedure for the generation and transport of all events in parallel and the storing of the resulting hit maps. From the hit map of event n and the drift of electrons in the distortion map of event $n-1$, the new space charges from primary ions and ion back-flow (20 ions per electron) are generated. This step can be parallelised if the distortion from the ion back-flow can be ignored.

8.2.4 Fast simulation

”Fast simulation” is fast full simulation and parametrised simulation or any mixture of both. Whereas analysis tries to undo the effects of efficiency and limited resolution, fast parametrised simulation applies these effects to simulated particle kinematics. Hence, an ideal place to integrate fast parametrised simulation is the existing ALICE analysis framework. The present implementation permits the combination of primary event simulation with the application of efficiency losses and 4-momentum smearing.

The implementation of a framework for fast full simulation or a combination of fast and full simulation can benefit from the existing VMC framework. Two new elements have to be added. The first is a dispatcher class that handles the step by step switching between different instances of the transport code applications, at least one of which would be a fast simulation package. The second is a virtual track class which is used by all transport engines to store current track information. The design resembles the ATLAS *Integrated Simulation Facility Approach* [6] with the added advantage that the ALICE fully

virtualised framework allows new components to be plugged in transparently.

Chapter 9

Data reduction

Data reduction is the driving requirement of synchronous reconstruction; the peak data rate from the read-out of the detectors is expected to be $\sim$1.1 TB/s. This implies the necessity for a substantial reduction of the data volume. Listed below are the data reduction procedures envisaged by different detectors.

TPC

The TPC is the detector with the largest data rate ($\sim$1 TB/s at 50 kHz Pb–Pb interaction rate). The reduction of its data is thus determining to a large degree the parameters of the processing model, i.e. the data rate to permanent storage. The envisioned steps for the TPC calibration have been already presented in the ALICE Upgrade LoI [1] and the TPC Upgrade TDR [2]. Table 9.1 lists these steps and the planned reduction factors.

Table 9.1: Data reduction steps for the TPC.

Data reduction step	Data reduction factor	Data rate at 50 kHz Pb–Pb interactions (GB/s)
Detector input	–	1000
Cluster finding	2.5	400
Cluster compression (cluster + track info.)	3	135
Background identification	2	68
Charge transformation & compression	1.35	50

The first step in the data reduction is the identification of the TPC clusters: the charge-deposits along the particle track through the TPC gas. This step is currently implemented in the ALICE High Level Trigger System and results in a reduction of the data volume by a factor 2.5. It has been successfully used in Run 1 and is implemented on the FPGA of the Read-out and Receiver Cards (H-RORC and C-RORC). This implementation is also the baseline for the upgraded TPC and will be implemented on the CRU.

The next step in the current system is an entropy-reducing transformation of some of the cluster properties using only information from the cluster and its neighbours in the data stream. The ordering of the clusters by the SAMPA or the FPGA algorithm allows the calculation of differences between subsequent clusters and thus benefits from correlations in the data. In this way, the effectiveness of lossless compression algorithms, e.g. Huffman coding, is increased significantly, allowing a further reduction of the cluster size by a factor of 1.6. Figure 9.1 shows the results obtained during Run 1. For the O^2 system these transformations including the lossless compression will be extended by calculating differentials for more cluster parameters. Some of these transformations require information from the track finding step for

the position and direction of the particle track at the cluster position. Using Run 1 data, the achievable reduction factor for the Huffman step has been estimated to be ∼3.

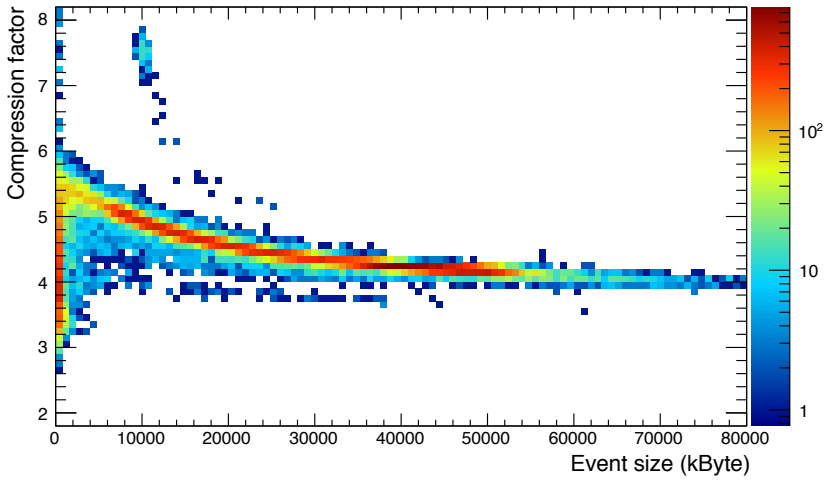

Figure 9.1: Data compression factor versus raw data size achieved using cluster finder and data-format optimizations in the High-Level Trigger during the Pb–Pb data taking in Run 1. Adopted from [3].

The availability of track information allows a further reduction of the TPC data size by removing clusters belonging to identified background tracks. Figure 9.2 shows a projection of the TPC clusters to the end-caps of the TPC. The clusters from low-momentum background tracks, like delta electrons, and from upstream ion-gas/beampipe interactions are clearly visible. By extending the momentum range of the current TPC tracking down to these low momenta, as well as allowing identification of tracks parallel to the beam-pipe, these background tracks and their associated clusters can be identified and subsequently removed from the data. Prototyping this algorithm on the TPC data of Run 1 has shown rejection factors in the range 2.5 (Pb–Pb collisions) to 4 (high intensity pp collisions), depending on the collision system and LHC beam intensity and background conditions. A conservative factor 2 has been assumed. A further reduction of the data volume is possible by transforming and/or removing the charge information from clusters on the track. This step makes use of the Landau distribution of the specific energy loss in the TPC gas and is estimated make a reduction of ∼1.35; it is conceptually very similar to the additional cluster transformations mentioned earlier.

ITS

At 50 kHz Pb–Pb, the ITS will ship ∼8.2 GB/s of data from hadronic collisions and QED interactions. The contribution from the noisy clusters is currently unknown, with a pessimistic upper limit of ∼28 GB/s (corresponding to a random noise probability of $\sim 10^{-5}$ per pixel). Therefore, together with ∼10% protocol overhead of ∼40 GB/s input data can be expected. The data reduction will start at the cluster finding stage on the FLPs, by storing the Huffman-coded relative values for the cluster centroid rows, chips as well as the identifier of the cluster topology rather than the absolute pixel addresses. With the multiplicity-dependent Huffman tables, it is estimated that the average cluster record size will be squeezed to ∼23 bits in the absence of noise and ∼19 bits if the noise dominates the data volume. This will lead to a compressed data rate of ∼26 GB/s for maximum noise scenario and ∼4.3 GB/s if the noise turns out to be negligible. In the pessimistic case of dominant noise a further reduction factor of ∼3 will be necessary.

If full synchronous ITS reconstruction proves to be feasible and efficient, discarding the clusters not attached to any track will eliminate virtually all noise clusters and approximately the half of real hit clusters (produced by non-reconstructable tracks), thereby reducing the peak data rate for storage to ∼3 GB/s. On the other hand, if fully efficient synchronous reconstruction appears to be too slow, discarding on the

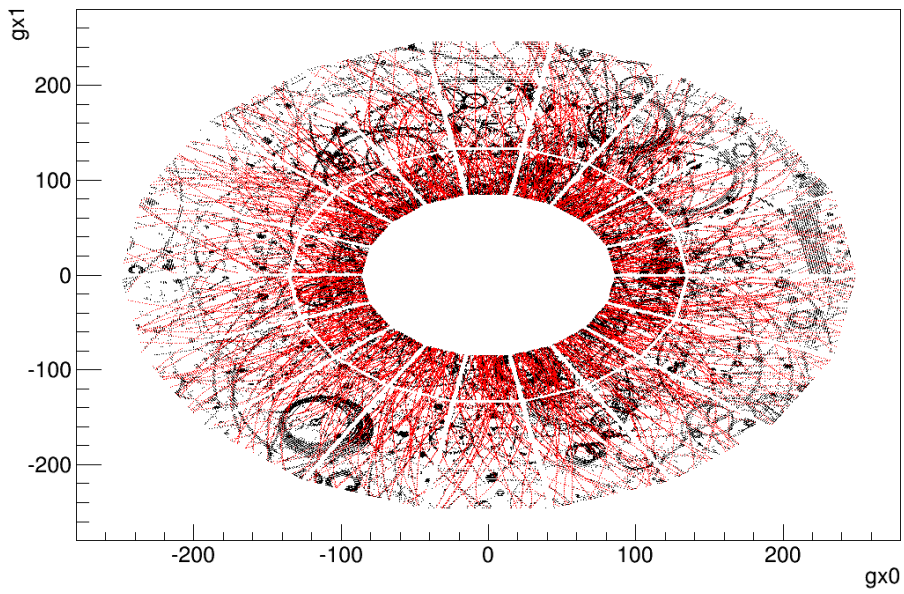

Figure 9.2: Projection of a pp event on the transverse plane of the TPC, showing clusters attached to physics tracks (red) and not assigned clusters (black).

outer layers only those clusters which can not be bound into any triplet losely converging to the luminous region will be considered.

TRD

The analysis of Run 1 data shows that only 10% of online tracklets can be matched to tracks extrapolated from TPC. The work is in progress to assess if this fraction remains the same for Runs 3 and 4 data. If this happens to be the case, we consider to store only those tracklets which are in vicinity of the extrapolation of TPC track found during synchronous reconstruction stage, which should lead to the reduction of the $\sim$20 GB/s input data rate to $\sim$3 GB/s data for permanent storage.

TOF

The optimisation of the current TOF data format is expected to give $\sim$20% reduction of the data rate of 2.5 GB/s (in Pb–Pb). This reduction will occur on the FLPs and depends on: the removal of slow control information (temperatures, thresholds) from the stored raw data which is available in the DCS; the suppression of the headers and trailers from TRM (TDC Read-out Module) data in case no hits are registered; improvements in the format of the data stored by DRM (Data Read-out Module). When running in continuous mode in pp, it should be noted that a larger data size (2.3 GB/s) and a lower impact of the data reduction technique is expected. The current plan is to use 10 μs windows for read-out. In this case, the number of TRMs without hits per asynchronous trigger will be lower.

MCH and MID

The input data raw rate from MCH extrapolated from Run 1 experience is $\sim$2.2 GB/s and can be compressed to $\sim$1.5 GB/s already on the FLPs. After pre-clustering on the FLPs and clusterisation on the EPNs, an output to permanent storage of $\sim$0.7 GB/s is expected once the pad information is dropped. The estimates are being made to check whether this operation is safe; without suppressing the pad information the estimated output would amount to $\sim$2.1 GB/s, in which case there is no gain from carrying out clusterisation in the synchronous mode.

MFT

Using the same type of sensors as the ITS, with the worst noise scenario of $\sim 10^{-5}$, the MFT will ship $\sim 7.2\,\mathrm{GB/s}$. For the data reduction, the same approach as for the ITS will apply, leading to $\sim 5.0\,\mathrm{GB/s}$ output to storage with Huffman compression only.

Calorimeters (EMC and PHS)

The EMC raw data at the rate of $\sim 2\,\mathrm{GB/s}$ and $\sim 4\,\mathrm{GB/s}$ from PHS can be reduced after the FLP processing. This can be achieved by compressing 30 10-bit samples on two floats for cell energy and time information.

Table 9.2 summarises the compression factors foreseen for every detector in routing data taking with Pb–Pb at 50 kHz interaction rate.

Table 9.2: Data rates for input to O^2 system and output to permanent storage for routine data taking with Pb–Pb at 50 kHz interaction rate.

Detector	Initial raw data rate (GB/s)	Compressed data rate (GB/s)	Data reduction factor
TPC	1000.0	50.0	20
ITS	40.0	26.0(8.0)	1.5(5)
TRD	20.0	3.0	6.6
TOF	2.5	2.0	1.25
MCH	2.2	0.7	2.9
MFT	10.0	5.0	2
EMC	4.0	1.0	4
PHS	2.0	0.5	4
Total	1080.7	88.2(70.2)	12.3(15.4)

Chapter 10

O^2 facility design

10.1 Introduction

The O^2 facility is the local computer farm at the ALICE experimental area where the O^2 architecture will be implemented, taking into account: the requirements presented in Chap. 3; the system architecture and design presented in Chap. 4 and Chap. 5; the needs of the software as presented in Chap. 7, 8, 9; the current state of the technology as presented in Chap. 6. Also discussed below are the infrastructure and the demonstrators.

10.2 O^2 Facility

This section describes the O^2 facility. Table 10.1 details the number of FLPs per detector. Table 10.2 details the roles and types of the computing nodes. Table 10.3 shows their requirements for network connectivity, and Tab. 10.4 summarises the storage needs, in terms of capacity and throughput.

The number of FLPs is derived from the number of detector links and associated read-out boards, balanced with the computing needs and FLP output network bandwidth specific for each detector. Each FLP will accommodate 1 or 2 PCIe read-out boards as explained in 6.2.1 and in 6.2.2. The O^2 facility will need about 250 FLPs in total.

Each FLP provides enough processing power to perform the initial data compression and memory to buffer the data while doing the Time Frame building.

The total output rates of FLPs assuming for simplicity that compression is only performed for TPC are:

$$totalFLPoutRate = DetectorDataRate/FLPcompression$$

$$= TPCrate/2.5 + otherDetectorsRate = 1000/2.5 + 100 = 500\,\mathrm{GB/s}$$

This makes an average output rate of $2\,\mathrm{GB/s}$ per FLP. The maximum rate is reached for the FLPs of the TPC inner chambers, which will each produce approximately $2.5\,\mathrm{GB/s}$ after the first compression.

The Time Frame rate is set to $50\,\mathrm{Hz}$ (as described in 5.2.2). The full Time Frame size on the EPN is:

$$fullTimeFrameSize = totalFLPoutRate/timeframeRate = 10\,\mathrm{GB}$$

The EPN incoming bandwidth is set to $10\,\mathrm{Gb/s}$. This is the expected commodity network port, selected to minimise the cost given the large number of EPNs.

$$EPNincomingBw = 10\,\mathrm{Gb/s}$$

Table 10.1: Number of read-out boards and FLPs per detector to O^2 system.

Detector	Number of read-out boards	Read-out board type	Number of FLPs
ACO	1	C-RORC	1
CPV	1	C-RORC	1
CTP	1	CRU	1
DCS	1	Network	1
EMC	4	C-RORC	2
FIT	1	C-RORC	1
HMP	4	C-RORC	2
ITS	23	CRU	23
MCH	25	CRU	13
MFT	14	CRU	7
MID	2	CRU	1
PHS	4	C-RORC	2
TOF	3	CRU	3
TPC	324	CRU	162
TRD	54	CRU	27
ZDC	1	CRU	1
Spares			2
Total			250

The minimum number of EPNs needed to read the data produced by the FLPs is:

$$minimumActiveEPNs = totalFLPoutRate/EPNincomingBW = 500 * 8/10 = 400$$

The worst case buffer size needed on the FLP for network transmission is:

$$minFLPbufferSize = EPNtimeToReceiveFullTimeframe * singleFLPoutgoingrate$$

$$= (fullTimeFrameSize/EPNincomingBw) * FLPoutgoingrate = 25\,\mathrm{GB}$$

Simulation indicated the additional memory needed for the read-out buffers and processing pipelines (see Fig. 10.6 and Fig. 10.7). Taking into account the Operating System and memory needs for the processes, an FLP system with 32 GB RAM is a realistic option.

The number of EPNs depends on the CPU processing needs for the Time Frame building and event reconstruction (for the final compression). In view of the requirements presented in Table 8.8, and assuming dual-CPU systems with 32 cores per CPU, the facility will need 1500 EPNs.

$$NEPNs = NCoresRequired/(NCpuPerNode * NCorePerCPU) = 100000/(2 * 32) \approx 1500$$

Each EPN will process one full Time Frame every

$$NEPNs/timeFrameRate = 1500/50 = 30\,\mathrm{s}$$

The memory needed on each EPN consists of: buffer to receive next time frame (10 GB), buffer for current time frame (10 GB), buffer for processing the current time frame (1 GB per CPU core) and 16 GB for the OS, i.e. a total of approximately 100 GB.

The aggregated bandwidth from all the EPN to the data storage amounts to:

$$detectorThroughput/compressionFactor = 1100/14 = 78.6\,\mathrm{GB}$$

It corresponds for each EPN to an average write disk throughput of $78.6/1500 = 52\,\mathrm{MB/s}$.

A total capacity of $90\,\mathrm{GB/s}$ is foreseen in order to have some headroom corresponding to a write disk throughput of $60\,\mathrm{MB/s}$ for each EPN.

The size of the Compressed Time Frame stored on the data storage is

$$CompressedTimeframeSize = detectorThroughput/(Timeframe Rate * compressionFactor) = 1100/(50*14) = 1.6$$

All the compressed data and the results of the reconstruction (CTF, ESD and AOD) are stored on disk. The yearly total amount of data to be stored in 2020 and 2021 is approximatively $54\,\mathrm{PB}$. A total capacity of $60\,\mathrm{PB}$ is needed for the data redundancy corresponding to $60000/1500 = 40\,\mathrm{TB}$ for each EPN. The total installed capacity will be of $68\,\mathrm{PB}$ in order to have some headroom.

Table 10.2: Nodes of the O^2 facility with their minimal characteristics.

Node Type	Number of nodes	Number of cores	Number and type of accelerators	Memory (GB)	Network bandwidth (Gb/s)	I/O slots PCIe Gen3
FLP TPC	162	2 x 6	FPGA/CRU	32	18	2
FLP ITS	23	2 x 24		32	12	1
FLP other detectors	65	2 x 6		32	12	1 or 2
EPN	1500	2 x 32	2 x GPU cards	100	10	0

Table 10.3: Network ports, throughput and bandwidth.

Node type	Network ports and bandwidth (Gb/s)	Data traffic type
FLP (out)	1 x 40GbE or 56 GbIB or 4 x 10GbE	Mostly outgoing traffic, continuous @ 10-20Gb/s
EPN (in)	1 x 40GbE or 56 GbIB or 1 x 10GbE	Input traffic in bursts (full speed during the burst and idle the rest of the time)

Table 10.4: Storage needs (throughput and capacity).

Storage location	Throughput MB/s	Capacity TB
EPN	60	40
Total	90,000	60,000

The least expensive network topology for TF building (the FLP to EPN data flow) is the layout 2 as presented in Fig. 5.4. It consists of 4 independent EPN clusters. An implementation example based on this approach is detailed below, with each FLP having four 10 Gb/s ports, each one being connected to one of the EPN clusters. The switching size of each cluster network is 250 FLPs at 10 Gb/s x 375 EPNs at 10 Gb/s, i.e. 632 ports in total. The aggregated outgoing traffic of all FLPs to one EPN cluster is $FLPoutgoingrate/4 = 1\,\mathrm{Tb/s}$.

This corresponds to an average output of 4 Gb/s per FLP. Each cluster, with at least 100 EPNs active at the same time (each reading 10 Gb/s) has to cope with the 1 Tb/s produced by FLPs. Using a leaf switch providing 96 ports at 10 Gb/s and 8 uplinks at 40 Gb/s, such as the DELL Z9000, the corresponding network can be built with following characteristics (per cluster):

- Each cluster has 7 switches, providing a total of 672 ports at 10 Gb/s (28 switches needed in total for O^2).
- Each switch is connected point-to-point to each other switch with a 40 Gb/s link, to implement a fully connected network mesh (21 cross-links).
- Each switch receives data from 37 FLPs (up to 259 FLPs in total).
- Each switch provides data to 54 EPNs (up to 378 EPNs in total).
- Each switch has 5 spare ports to accommodate more nodes if needed.
- In these conditions, if the Time Frames are distributed evenly across EPNs (one switch after the other), the uplink traffic of each switch is symmetric and amounts in each direction to $37*4*6/7 = 125\,\mathrm{Gb/s}$, i.e. less than 50% of the available bandwidth ($7*40 = 280\,\mathrm{Gb/s}$).

A second topology for TF building is the layout 3 as presented in Fig. 5.5. An implementation combining the Infiniband and Ethernet technologies would for example consist of:

- 10 FLP edge switches (Mellanox SX6025) with 36 IB ports at 56 Gb/s, 25 as input for FLPs and 10 as output to a director switch.
- 1 director switch (Mellanox SX6512) with 216 IB ports at 56 Gb/s, 50 as input for the FLP edge switches and 50 as output to the SEPNs.
- 50 SEPNs performing the TF building and distributing the data to 50 independent EPN clusters through a switch (Mellanox SX 1024) with 4 Ethernet ports at 40 Gb/s from the SEPNs and 30 Ethernet ports at 10 Gb/s to the EPNs.

10.3 Power and cooling facilities

The O^2 facility will be located at the LHC Point 2 and will require a modification of the facilities available at the ALICE experimental area. Two Counting Rooms (CR1 and CR2) are available with 40 racks in each with a usable height of 40 U in CR1 and 45 U in CR2.

CR1 has been entirely rejuvenated with new racks and will be used for the racks containing FLPs and services. CR2 would not be sufficient for the racks containing EPNs and the data storage equipment. Moreover, the racks would have to be exchanged to increase their cooling facility. A new room (CR0) will be installed on the surface for the EPNs and the data storage equipment. Table 10.5 summarizes the power and cooling needs of the O^2 facility assuming usual power consumption and dissipation.

Table 10.5: Location, rack space, power and cooling needs of the O^2 facility.

Type	Number of items	Item height (U)	Location-Total height (U)	Number of Racks	Power per rack (kVA)	Total power (kVA)	Cooling per rack (kW)	Total cooling (kW)
FLP	250	2	CR1 - 35	18	12	216	14	252
EPN	1500	1	CR0 - 54	34	50	1700	50	1700
SEPN	50	1	CR0 - 54	1	12	12	12	12
Storage	34	9	CR0 - 54	7	50	350	50	350
Network								
Dataflow	10	3	CR1 - 40	2	12	24	12	24
Dataflow	15	3	CR0 - 54	2	12	24	12	24
Control	4	1	CR1 - 40	1	12	12	12	12
Control	14	1	CR0 - 54	1	12	12	12	12
Services	110	1	CR1 - 40	4	12	48	12	48
Total			CR1	25		300		336
			CR0	45		2086		2086
Grand total				70		2398		2434

The existing power distribution at Point 2 includes a total UPS power of 500kVA available in CR1 and CR2 plus 270kW of normal power for each CR. The power distribution requires a moderate upgrade to meet the requirements of the O^2 facility. To address the huge cooling needs, the possibility of using mixed water or free air cooling are considered.

10.4 Demonstrators

This section is dedicated to the demonstrators, proof of concepts or prototypes which have been developed to show key elements of the O^2 system.

10.4.1 Data transport

Two different systems were used for the performance measurement of data transport layer in ALFA. The performance tools delivered by ZeroMQ were also used to investigate any penalties introduced by FairMQ package.

Ethernet-based prototype

This system consists of 8 dual-Xeon machines, 4 connected with 40 Gb Ethernet while the other 4 are connected with 10 Gb Ethernet. Using a message part size of 10 MB and sending it from the FLPs to the EPN prototypes, a rate of about 37.6 Gb/s was achieved using 4 core CPUs for sending data (see Fig.10.1). It demonstrates that the overhead introduced by the FairMQ and ZeroMQ is marginal with a bandwidth equivalent to 94% of the maximum and that the technology scales well above the performance required by the FLPs on their output network link.

Infiniband-based prototype

The second system is composed of a 40 Gb/s IB using 4 dual-Xeon machines (Intel Xeon E5520 with 4 physical cores and 8 threads each) all running the same software but with the IPoIB protocol. Three

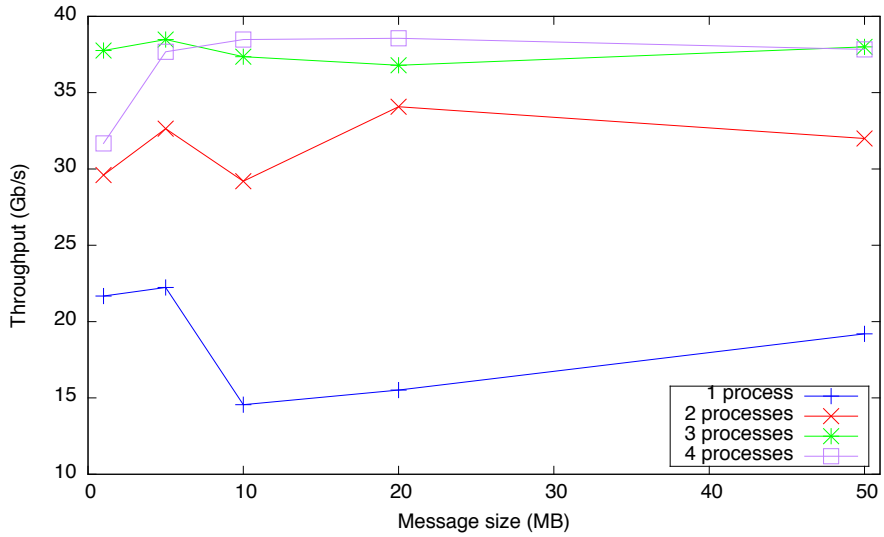

Figure 10.1: Throughput between two machines connected with 40 Gb/s Ethernet.

processes were used to send data from one machine and 4 processes on each of the other machines received data. A message size of 10 MB was used. An average rate of 2.5 GB/s was reached without any optimisation of the kernel parameters. This test confirms that the marginal overhead introduced by the FairMQ and ZeroMQ software with a measured performance equivalent to the one measured with benchmarking programs (see Tab. 6.6). The test also demonstrates the portability of the FairMQ software to different network technologies (Ethernet and IB) which provides the independence about the underlying network technology.

10.4.2 The Dynamic Deployment System

The Dynamic Deployment System (DDS) is an independent set of utilities and interfaces, providing a dynamic distribution of different user processes for any given topology on any Resource Management System (RMS). The DDS uses a plug-in system in order to deploy different job submission front-ends. The first and the main plug-in of the system is a Secure Shell (SSH) that can be used to dynamically transform a set of machines into user worker nodes. The DDS functions are the following:

- Deploy a task or set of tasks
- Use any RMS (Slurm, Grid Engine, ...etc)
- Execute nodes securely (watchdog)
- Support different topologies and task dependencies
- Support a central log engine

During 2014, the core modules of the DDS were developed and the first stable prototype was released. This has been tested on the ALICE HLT development cluster using 40 computing nodes with 32 processes per node. The SSH plugin for DDS has been used to successfully distribute and manage 1280 ALICE O^2 user tasks (640 FLPs and 640 EPNs). The DDS was able to propagate the allocated ports for each process to the dependent processes and set the required topology for the test. Throughout the test on this cluster, one DDS commander server propagated more than 1.5 million properties in less than 5 seconds.

A test with more nodes and cores is ongoing at the test cluster of the new GSI cluster (Kronos). A rack with 50 dual-Xeon machines (Intel Xeon E5-2660 with 10 physical cores and 20 threads each) is being used for the test. On this cluster there are more than 10000 ALFA devices (FLPs, EPNs, and one Sampler). During this second test, DDS has propagated about 77 millions key-value properties. The start-up time of the whole deployment by the DDS was 207 seconds for the propagation of all required properties and devices, the bind/connect and enter into RUN state. This test was an important milestone for the DDS, demonstrating a scalability adequate for the O^2 system. The target is to scale up to 100 k devices and reduce the start-up time for the full deployment down to 10-50 seconds (depending on the number of properties the system needs to propagate).

10.4.3 Future demonstrators

Several other demonstrators are under design.

A FLP demonstrator is being developed. It will consist of all the software needed to read out one CRU and make some quality control over the data. This demonstrator is the base of the system which will be used by the new or upgraded detectors during their test phase.

A global demonstrator is also scheduled. It will implement the whole data processing chain and will use some TF assembled from existing real data collected during Run 1.

10.5 Computing system simulation

The computing system simulation has been done with Omnet++ [1] and the Monarc simulation tool [2]. See Sec. C.2.6 for more details about this choice.

The high-level simulation based on the Monarc tool includes all the elements to be simulated in a large program. It has been used to simulate the network and the buffering needs as reported in Sec. 10.5.1 and Sec. 10.5.2.

The Omnet++ simulation models are made of several blocks or modules interacting with each other by sending messages through links. A number of modules has been customised, while others were already available in Omnet++. Different simulation models have been made to reflect the levels of detail needed, while the same parts of the real system are represented by different modules in the simulation. For instance, an FLP in a simulation of the network between FLPs and EPNs will be a data source, whereas in a simulation that is more focused on the part of the system before the FLPs, it will be a data sink. In a complete simulation scenario it will be simulated using yet another module. Three different models (network, full system and storage) have been created and used to evaluate the overall system scalability (see Sec. 10.5.4 and Sec. 10.5.3).

10.5.1 Simulation of the network needs

The study of the data traffic pattern is essential to properly dimension the system. Two different network layouts have been simulated: the layouts 2 and 3 introduced in Chap. 5 and shown in Fig. 5.4 and Fig. 5.5. Two types of FLPs are used for these simulations: the TPC FLPs transferring an aggregate of 400 GB/s to the EPNs and the FLPs of the Other Detectors (OD FLPs).

In order to maintain the total data throughput, several transfers must be performed in parallel by each FLP. All the FLPs transfer data all the time to some EPNs. The EPNs receive a burst of data when being selected as a destination and nothing when processing those data. Figure 10.2 shows the network bandwidth out of the TPC and OD FLPs and in the EPNs for the layout 2 with 100 or 20 maximum number of concurrent transfers out of the FLPs (left-hand and right-hand panels respectively). In the latter case, there is an initial perturbation before the system settles in a state similar to the first case but with much less parallel data transfers.

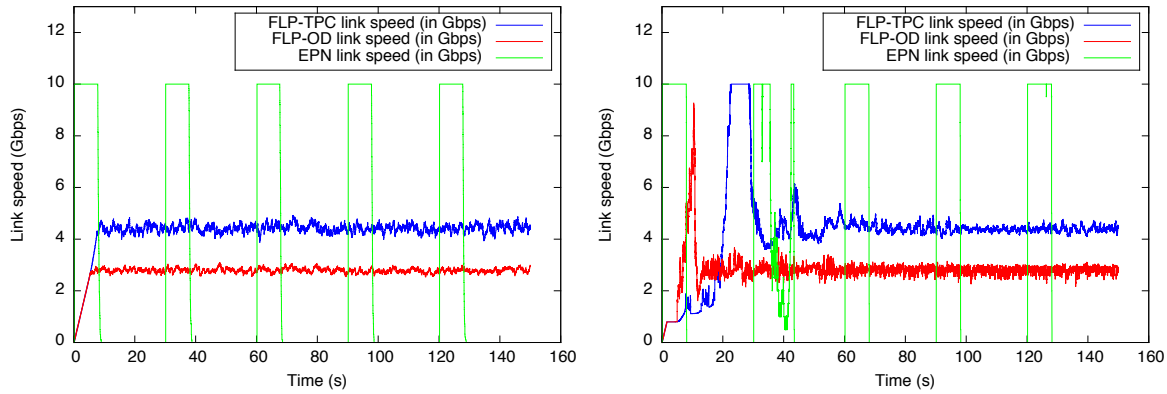

Figure 10.2: Link speed on the FLPs and EPNs for a network layout with four EPN subfarms for 100 (left) and 20 (right) parallel transfers from the FLPs.

For the layout 3, the FLPs transmit the data to SEPNs in charge of buffering the data and redistributing it to the EPNs. The network bandwidth out of the TPC and OD FLPs and in the SEPNs for the layout 3 is shown in Fig. 10.3 for a configuration based on an Infiniband network at 56 Gb/s. As can be seen the network is well used with the SEPN receiving data at the nominal bandwidth more than 50% of the time.

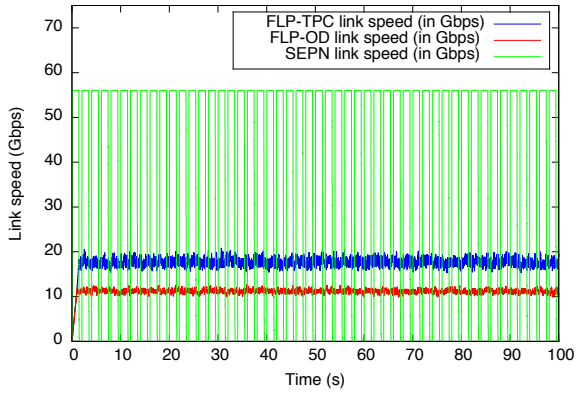

Figure 10.3: Link speed on the FLPs and SEPNs for a network layout with SEPNs.

This simulation has also been used to verify the bisection data traffic in the system. Figure 10.4 shows the data traffic for one of the 4 EPNs subfarms of layout 2 equal to 1 Tb/s for the two conditions simulated above: 100 or 20 maximum number of concurrent transfers out of the FLPs (left-hand and right-hand panels respectively).

For the layout 3, the bisection data traffic of 4 Tb/s can be seen in Fig. 10.5.

10.5.2 Simulation of the buffering needs

The data buffering for the different types of nodes have also been estimated using the high-level simulation. Two kinds of buffer management have been simulated. Either no special management is used and the space made available by the transfer of a STF from the FLP is returned to the pool of available memory at the end of the transfer or the space is returned gradually to the pool during the transfer. Given the large size of the STF, it is relevant to see whether this special management would provide any substantial benefit.

Figure 10.6 shows the buffering need for the TPC and OD FLPs of layout 2 with and without buffer management for the two conditions simulated above: 100 or 20 maximum number of concurrent transfers

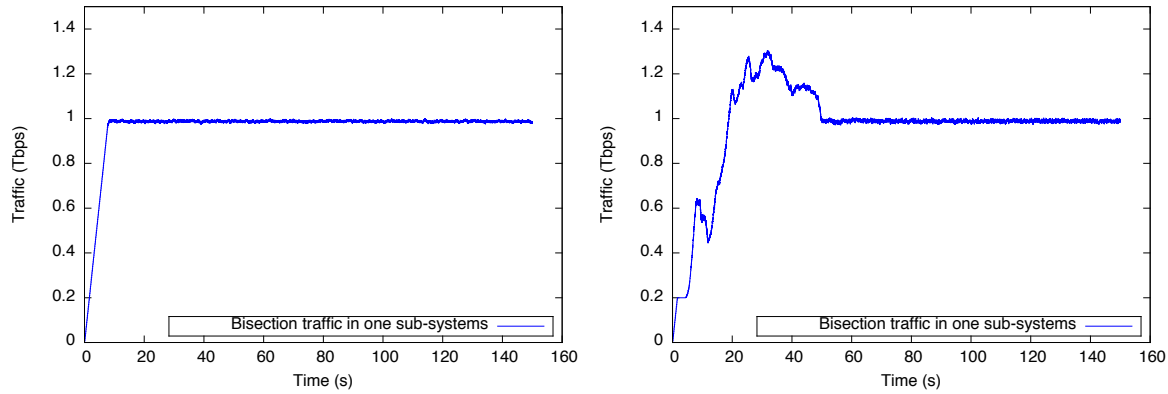

Figure 10.4: Bisection traffic in one sub-system for a network layout with four EPN subfarms for 100 (left) and 20 (right) parallel transfers from the FLPs.

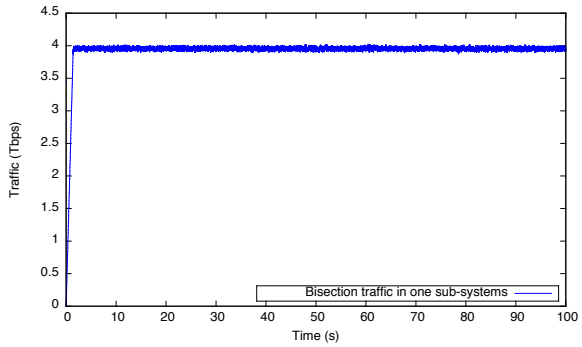

Figure 10.5: Bisection traffic in the network layout with SEPNs.

out of the FLPs (left-hand and right-hand panels respectively).

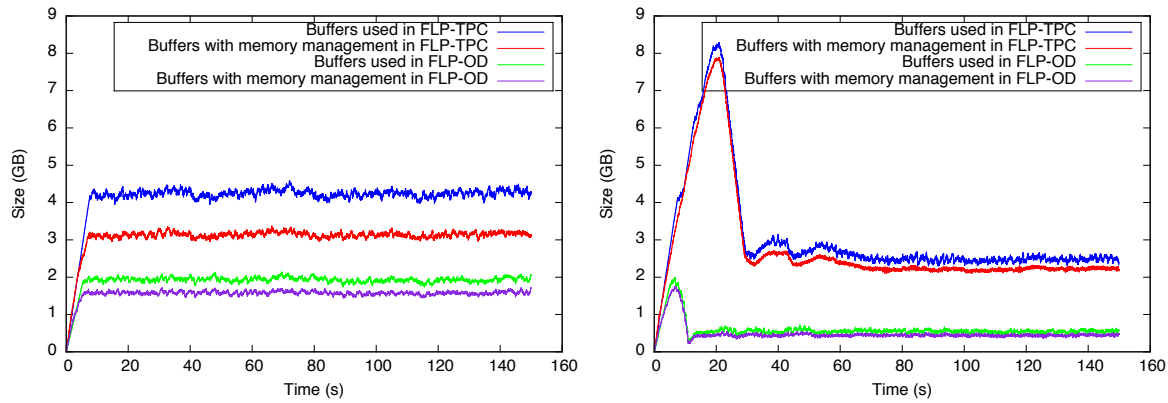

Figure 10.6: Size of the memory buffers with and without memory management for a network layout with four EPN subfarms for 100 (left) and 20 (right) parallel transfers from the FLPs.

Figure 10.7 shows the buffering need in the TPC and OD FLPs of layout 3 with and without buffer management.

10.5.3 Data storage simulation

The simulation has also allowed to evaluate the total size of the data that will be stored during a typical Pb–Pb run. Figure 10.8 shows the total amount of data stored in the O^2 facility as a function of time culminating to a total amount of about 30 PB. A realistic input data pattern has been generated based on the length of Pb–Pb runs in 2011 and the data size and reduction for Run 3.

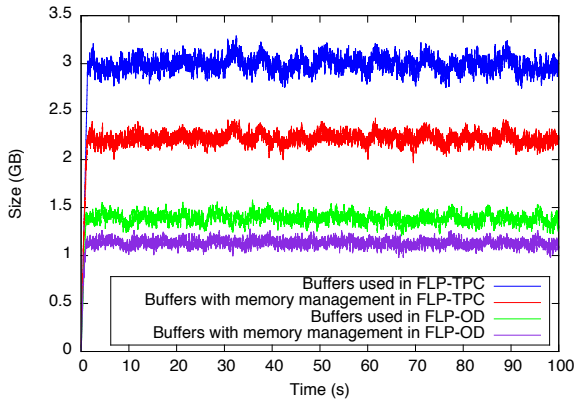

Figure 10.7: Size of the memory buffers with and without memory management for a network layout with SEPNs.

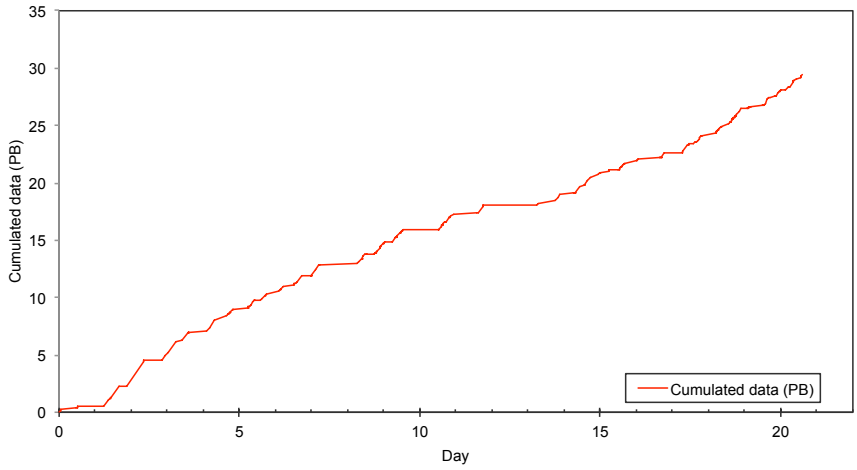

Figure 10.8: Total data stored for a month of data taking similar to Run 1.

10.5.4 Simulation of the system scalability

The O^2 system must scale from its nominal interaction rate of 50 kHz up to 100 kHz. The simulation has been used to verify its scalability and the frequency at which the system does not scale anymore. The scalability has been estimated by simulating the total latency of the TF in the system. This latency is stable as long as the system is in its scalability interval and increases out of this interval.

Figure 10.9 shows that the network layout 2 with a 40 Gb/s Ethernet network scales well till 80 kHz but is overloaded at 90 kHz which is below the target scalability of the O^2 system.

Figure 10.10 shows that the network layout 3 with a 56 Gb/s IB network scales well till 140 kHz which is well above the target scalability of the O^2 system. The better performance of the network layout 3 is normal given the higher network bandwidth. The layout 2 can only be implemented with 40 Gb/s Ethernet network links which can be split into 4 independent 10 Gb/s Ethernet network links and not with the 56 Gb/s IB network link. The layout 3 with a 56 Gb/s IB network has therefore been used for the evaluation of the project cost.

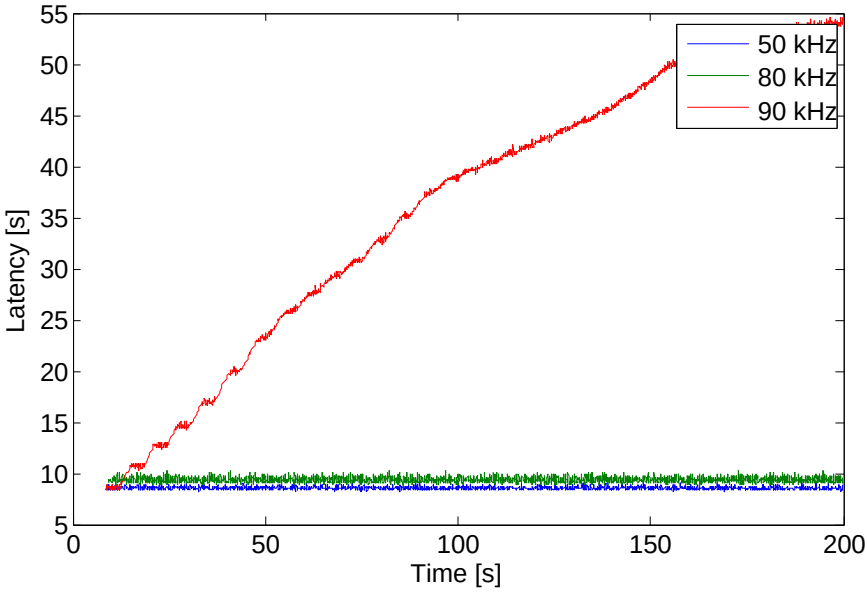

Figure 10.9: System scalability of the network layout 2 at up to 80 kHz.

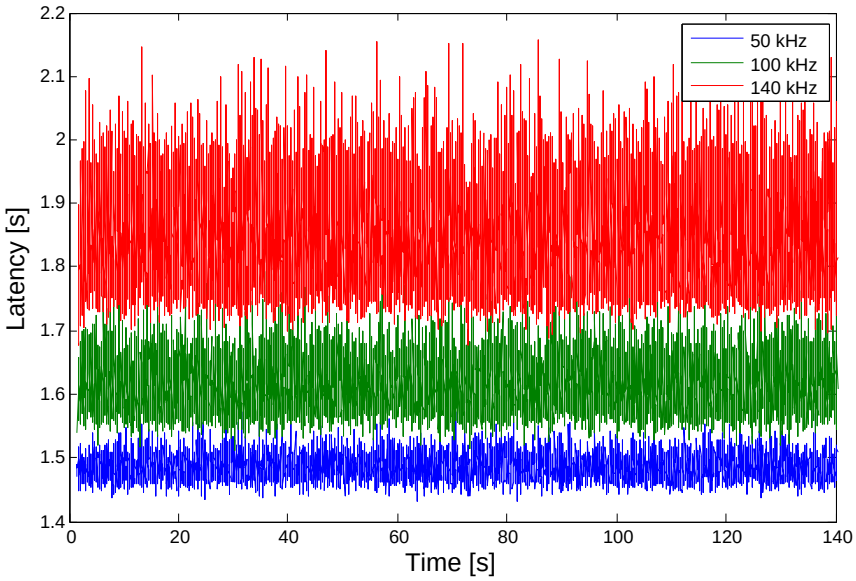

Figure 10.10: System scalability of the network layout 3 at up to 140 kHz.

Chapter 11

Project organisation, cost estimate and schedule

11.1 Project management

The O^2 project is run by the present leaders of the DAQ, HLT and Offline projects. They are also members of the O^2 Steering Board (SB), the body that deals with managerial, financial, technical and organisational matters.

The R&D work is being carried out by the O^2 Computing Working Groups (CWGs), listed in Tab. 11.1. The CWGs include members from the DAQ, HLT and Offline Projects and other institutes who have joined the O^2 project.

Table 11.1: O^2 Project Computing Working Groups and their topics.

Group	Topic	Group	Topic
CWG 1	Architecture	CWG 8	Physics Simulation
CWG 2	Tools & Procedures	CWG 9	Quality Control, Visualization
CWG 3	Data flow	CWG 10	Control, Configuration, Monitoring
CWG 4	Data Model	CWG 11	Software Lifecycle
CWG 5	Computing Platforms	CWG 12	Hardware
CWG 6	Calibration	CWG 13	Software framework
CWG 7	Reconstruction		

11.2 O^2 Project

The O^2 Project holds regular plenary meetings during ALICE weeks and mini-weeks. These meetings are open to all members of the ALICE collaboration. Plenary meetings provide a forum for discussion and presentation of major topics on the design, construction and operation of the O^2 system as well as for regular reports from the CWGs .

The institutes are represented by their team leaders in the O^2 Institute Board (IB). The O^2 project leaders are ex-officio members of the IB. The managerial, financial and organisational issues are discussed and decided on in the IB. This board also endorses technical matters recommended by the CWGs and proposed by the SB.

All institutes participating in the O^2 Project are shown in Tab. 11.2.

Table 11.2: Institutes participating in the O^2 Project.

	Country	City	Institute	Acronym
1	Brasil	São Paulo	University of São Paulo	USP
2	CERN	Geneva	European Organization for Nuclear Research	CERN
3	Croatia	Split	Technical University of Split	FESB
4	Czech Republic	Rez u Prahy	Nuclear Physics Institute, Academy of Sciences of the Czech Republic	ASCR
5	France	Clermont -Ferrand	Laboratoire de Physique Corpusculaire (LPC), Université Blaise Pascal Clermont-Ferrand, CNRS-IN2P3	LPC
6	France	Grenoble	Laboratoire de Physique Subatomique et de Cosmologie (LPSC), Université Grenoble-Alpes, CNRS-IN2P3	LPSC
7	France	Nantes	SUBATECH, Ecole des Mines de Nantes, Université de Nantes, CNRS-IN2P3	SUBATECH
8	France	Orsay	Institut de Physique Nucléaire (IPNO), Université Paris-Sud, CNRS-IN2P3	IPNO
9	France	Strasbourg	Institut Pluridisciplinaire Hubert Curien	IPHC
10	Germany	Darmstadt	Research Division and ExtreMe Matter Institute EMMI, GSI Helmholtzzentrum für Schwerionenforschung	GSI
11	Germany	Frankfurt	Frankfurt Institute for Advanced Studies, Johann Wolfgang Goethe-Universität	FIAS
12	Germany	Frankfurt	Institut für Informatik, Johann Wolfgang Goethe-Universität Frankfurt	IRI
13	Hungary	Budapest	Wigner RCP Hungarian Academy of Sciences	WRCP
14	India	Jammu	University of Jammu	JU
15	India	Mumbai	Indian Institute of Technology	IIT
16	Indonesia	Bandung	Indonesian Institute of Sciences	LIPI
17	Korea	Daejeon	Korea Institute of Science and Technology Information	KISTI
18	Korea	Sejong City	Korea University	KU
19	Poland	Warsaw	Warsaw University of Technology	WUT
20	Romania	Bucharest	Institute of Space Science	ISS
21	South Africa	Cape Town	University of Cape Town	UCT
22	Thailand	Bangkok	King Mongkut's University of Technology Thonburi	KMUTT
23	Thailand	Bangkok	Thammasat University	TU
24	Turkey	Konya	KTO Karatay University	KTO
25	United States	Berkeley, CA	Lawrence Berkeley National Laboratory	LBNL
26	United States	Detroit, MI	Wayne State University	WSU
27	United States	Houston, TX	University of Houston	UH
28	United States	Knoxville, TN	University of Tennessee	UTK
29	United States	Oak Ridge, TN	Oak Ridge National Laboratory	ORNL
30	United States	Omaha, NE	Creighton University	CU
31	United States	Pasadena, CA	California Institute of Technology	CALTECH

11.3 Schedule

The ALICE upgrade is planned to be operational after LS2 continuing into the High Luminosity LHC (HL-LHC) era after LS3 in 2025. According to the current LHC schedule, LS2 will take place in 2018-19 and LS3 will start in 2023. As for Run 1, the O^2 software framework will be released early and often and the O^2 facility will be incrementally deployed to give optimal support to the detector upgrade activities. A first version of the software framework, including that for detector read-out, will be released at the start of 2017 in time for the first tests with the ITS, MFT and TPC electronics. Significant fractions of the facility will be available for the global commissioning of the ITS, MFT and TPC detectors towards the end of 2018 before these detectors are installed in the experimental area. The system deployed in 2019 will handle the read-out of all detectors and the data processing at a reduced rate. The deployment of the full data processing capacity will continue in 2020 following a schedule compatible with the LHC and the beginning of the heavy ion collisions. The main project activities and milestones are shown in the Gantt chart shown in Fig. 11.1.

11.4 Responsibilities and human resources estimates

The work on the design and development of the O^2 system has been distributed among the participating Institutes. A summary of the responsibilities is shown in Tab. 11.3 for the period from 2015 to 2019.

Table 11.3: Sharing of responsibilities and human resources needed.

Tasks	Institutes	Human resources (FTE)
Architecture	CERN, FIAS,GSI, IRI	2.0
Tools, procedure and software process	CERN, IPNO, JU, LIPI, WRCP	2.0
Data flow, detector read-out	CALTECH, CERN, FESB, FIAS, IRI, LIPI, WRCP	16.0
Computing platforms	CERN, FIAS, IRI, JU, KISTI, KMUTT, KU, ORNL	9.0
Software framework and data model	CERN, IPNO, GSI, LBNL	14.0
Calibration	JU, WSU	29.0
Reconstruction	CERN, FESB, GSI, IPHC, LIPI, LPC, SUBATECH, UH, WSU	44.0
Physics simulation	CERN, CU, IPHC, IPNO, LBNL, ORNL, UH, UTK	32.0
Data quality monitoring and visualisation	CERN, ISS, JU, WUT	14.5
Control, configuration, monitoring and logging	ASCR, CALTECH, CERN, CU, KMUTT, IRI	13.0
O^2 facility hardware procurement, installation	CERN, FIAS, IRI, GSI	7.7
O^2 facility and grid/cloud operation	CERN, KISTI	6.3 and M&O

The manpower estimate for software tasks closely related to detectors includes the design and development of the core software and the integration and commissioning of detector specific algorithms but not the design and development of the algorithms themselves.

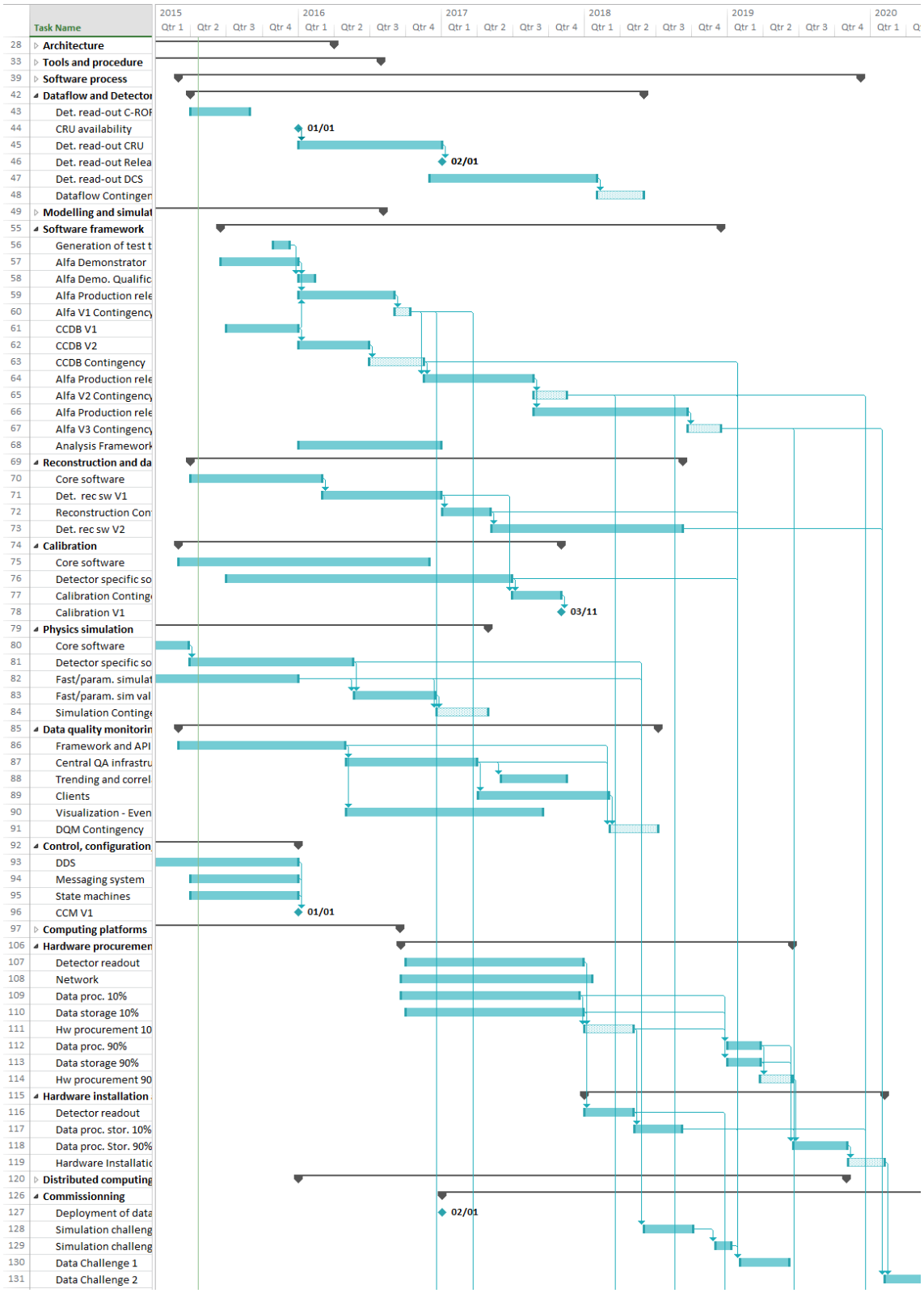

Figure 11.1: O^2 Project schedule.

11.5 Cost estimates and spending profile

The project cost estimate is given in Tab. 11.4. All CORE costs are included: material cost of the equipment purchased or produced; the industrial or outsourced manpower for production and general infrastructure at the experimental area. The institutes' personnel costs and the R&D activities are not included.

Table 11.4: Cost estimates and spending profile.

Item	Cost estimate (kCHF)	Spending profile 2018 (kCHF)	Spending profile 2019 (kCHF)
Infrastructure	776	465	310
FLPs and CRUs	916	550	366
EPNs	5,152	515	4,636
Data storage	2'168	217	1,951
Network	1,018	509	509
Servers	438	306	131
Total	10,467	2,563	7,905

The maintenance and operation costs are not included in Tab. 11.4, but will be funded by the ALICE Maintenance and Operation (M&O) budget as during Run 1 and 2.

11.6 Production and procurement

The O^2 facility will be almost entirely comprised of commercial off-the-shelf equipment (servers, PCs, data storage equipment, network switches and routers). This equipment will be procured by one of the institutes involved in the project according to the calendar specified in Tab. 11.4. None of the equipment is from a single source so that the procurement can be done following the standard procedure for a competitive tender.

The CRU used to read out the detectors and perform the TPC cluster finder in its FPGA is the only piece of equipment which might not be commercial off-the-shelf. The CRUs will be financed by the detector groups and the O^2 project. They will be acquired centrally by the collaboration. The baseline solution for the CRU is the PCIe 40 board designed by the LHCb collaboration. The full specification will be made in collaboration with LHCb and the hardware purchased after a competitive tender.

11.7 Risk analysis

The project risks have been taken into consideration and addressed in the following paragraphs.

All the hardware and software technologies used for the O^2 system exist already and have been demonstrated to work according to requirements. None of the hardware items under consideration comes from a single source. Every piece of equipment used today for the tests could be replaced by one of another brand at the time of the purchase. The technologies employed do not constitute risks for the project.

The budget of the project has been evaluated on the basis of recent purchases by ALICE or the CERN IT department and on conservative projections of the price evolution. Still there might be variations due to independent factors. The budget estimate includes a margin to meet this risk. In the case where a component would be more expensive than anticipated, then its deployment would be staged.

Exchange rate risk: the project's budget is in Swiss Francs but a large fraction of the contributions is made in Euros. The project's expenses mostly concern computing equipment the prices of which are initially expressed in US Dollars. Variations of the exchange rate between these three currencies can affect the budget of the project.

If a risk related to the budget is confirmed, the deficit will be covered by increasing the fraction of data sent to the Tier 1 sites for asynchronous processing and the usage of parking for data processed during LS3.

The project schedule has been evaluated taking into account all the experience accumulated by the different teams contributing to the project. The hardware procurement and deployment has been scheduled with sufficient margin to absorb delays due to the administrative procedure or in the delivery of goods. Software development is often a difficult process to track. The strategy adopted in this project is to release early and often in order to have a realistic feedback on the progress. Margins have been introduced for all the major deliverables to address the unforeseen delays.

11.8 Data preservation

A Study Group on Data Preservation and Long Term Analysis in High Energy Physics (DPHEP) was established in 2009, involving all of the main HEP institutes and experiments worldwide. In 2012, the group brought out a Blueprint document [1]. A summary of this publication was used in the update of the European Strategy for Particle Physics. The revised version of the strategy was officially adopted at the 16th European Strategy Session of the CERN Council in Brussels in May 2013. Data Preservation is one of the few computing and infrastructure related items included in the document.

To be cost effective, long term data preservation must be fully integrated into the ongoing activities of the experiments and, in the context of WLCG, the project itself. To be able to retain the full potential of the data over decades is not a trivial matter, nor is it guaranteed to succeed, as past experience shows. Some changes in culture and practice may be required to reach the desired goals effectively. These changes imply designing the software, the complete environment including the framework as well as the necessary validation steps in such a way as to allow future use of the data. The DPHEP Blueprint foresees that common projects should include the use of virtualisation techniques, validation frameworks and collaboration with different projects. This last point is directed specifically at ongoing work on the redesign and implementation of software for concurrency and modern architectures.

Documentation, long term data preservation at various levels of abstraction, data access, analysis policy and software availability are the key elements of such a data preservation strategy that allow future collaborators, the wider scientific community and the general public to analyse data for educational purposes and for eventual reassessment of the published results.

11.9 Open access

In 2013 the ALICE Collaboration Board approved a Data Preservation Strategy document, which is in line with similar documents already drafted by other experiments and stipulates the principle of open access to the data, software and documentation.

The ALICE collaboration is in agreement with the principle of open access to data, software and documentation, that will allow the processing of data by non-ALICE members under the conditions listed in the future policy implementation document. Data with high abstraction, such as AODs, will be conditionally made public after 5 years for 10% of the data and after 10 years for 100% of the data. Depending on the available resources for open access, external users can be conditionally granted access to computing resources to process data. Different levels for preservation and open access are currently foreseen

and described in the ALICE policy document.

11.10 O^2 TDR editorial committee

The composition of the O^2 editorial committee is: L. Betev, P. Buncic, S. Chapeland, F. Cliff, P. Hristov, T. Kollegger, M. Krzewicki, K. Read, J. Thäder, P. Vande Vyvre, B. von Haller.

11.11 O^2 TDR authors

The following people have contributed to the work presented in this TDR (The arabic institute numbers refer to the list from Tab. 11.2 and the roman institute numbers refer to footnotes):

T. Achalakul[22], D. Adamova[4], N. Agrawal[15], K. Akkarajitsakul[22], A. Alarcon Do Passo Suaide[1], T. Alt[11], M. Al-Turany[10], C. Alves Garcia Prado[1], Ananya[15], L. Aphecetche[7], A. Avasthi[15], M. Bach[11], S. Bae[17], R. Bala[14], G.G. Barnaföldi[13], A. Bhasin[14], J. Belikov[9], F. Bellini[i], L. Betev[2], L. Bianchi[27], J. Blomer[2], T. Breitner[11, 12], P. Buncic[2], F. Carena[2], W. Carena[2], B. Changaival[22], S. Chapeland[2], M. Cherney[30], V. Chibante Barroso[2], H. Cho[18], P. Chochula[2], F. Cliff[ii], F. Costa[2], P. Crochet[5], L. Cunqueiro Mendez[2], S. Dash[15], E. David[13], C. Delort[2], E. Dénes[13], T. Dietel[21], R. Divià[2], B. Doenigus[10], H. Engel[12], D. Eschweiler[11], U. Fuchs[2], A. Gheata[2], M. Gheata[20], A. Gomez Ramirez[12], S. Gotovac[3], S. Gorbunov[11], L. Graczykowski[19], A. Grigoras[2], C. Grigoras[2], A. Grigore[2], R. Grosso[27], R. Guernane[6], A. Gupta[14], N. Hadi Lestriandoko[16], F. Hensan Muttaqien[16], I. Hřivnáčová[8], P. Hristov[2], C. Ionita[2], M. Ivanov[10], M. Janik[19], H. Jang[17], P. Jenviriyakul [22], S. Kalcher[11], N. Kassalias[1], U. Kebschull[12], R. Khandelwal[15], S. Kushpil[4], I. Kisel[11], G. Kiss[13], T. Kiss[13, iii], T. Kollegger[10], M. Kowalski[iv], M. Kretz[11], M. Krzewicki[11], I. Kulakov[12], Laosooksathit[29], C. Lara[12], A. Lebedev[10], I. Legrand[31], V. Lindenstruth[11], A. Maevskaya[v], P. Malzacher[10], A. Manafov[10], A. Morsch[2], E. Mudnic[3], S. Na Ranong [22], B. Nandi[15], M. Niculescu[20], J. Niedziela[19], J. Otwinowski[vi], J. Ozegovic[3], V. Papi[3], S. Park[17], A. Peters[2], P. Phunchongharn[22], P. Pillot[7], M. Planinic[vii], J. Pluta[19], N. Poljak [vii] S. Prom-on[22], K. Pugdeethosapol [22], S. Pumma[22], A. Rabalchenko[10], S. Rajput[14], K. Read[28], A. Ribon[2], M. Richter[viii], D. Rohr[11], D. Rosiyadi[16], G. Rubin[13], R. Sadikin[16], R. Shahoyan[2], A. Sharma[14], G. Simonetti[2], O. Smorholm [ii] C. Soós[2], S. Sumowidagdo[16], J. Sun[18], M. Szymanski[19], A. Telesca[2], J. Thäder[25], A. Timmins[27], A. Udupa[15], P. Vande Vyvre[2], F. Vennedey[11, 12], L. Vickovic[3], B. von Haller[2], S. Wenzel[2], N. Winckler[10], T. Wirahman[16], K. Yamnual [22], C. Zampolli [i], and M. Zyzak[12].

[i] University and Sezione INFN, Bologna, Italy
[ii] School of Physics and Astronomy, University of Birmingham, Birmingham, United Kingdom
[iii] Cerntech Ltd., Budapest, Hungary
[iv] The Henryk Niewodniczanski Institute of Nuclear Physics, Polish Academy of Sciences, Cracow, Poland
[v] Institute for Nuclear Research, Academy of Sciences, Moscow, Russia
[vi] Polish Academy of Sciences, Cracow, Poland
[vii] Institute Rudjer Boskovic, Zagreb, Croatia
[viii] Department of Physics, University of Oslo, Oslo, Norway

Appendices

Appendix A

Glossary

Acronyms

A

AF	Analysis Facility
ALICE	A Large Ion Collider Experiment
AliROOT	ALICE software framework based on ROOT
AOD	Data type: Analysis Object Data
API	Application Programming Interface
APU	Accelerated Processing Unit combining a CPU and a GPU
ASCII	American Standard Code for Information Interchange

B

BC	Bunch Crossing

C

CAF	Central Analysis Facility
CCDB	Condition and Calibration Data Base
CCM	Control, Configuration and Monitoring
CDH	Common Data Header
CEPH	Storage platform
CERNVMFS	CERN Virtual Machine File System
CFS	Cluster File System
CPU	Central Processing Unit
CR0	New Counting Room needed for the O^2 project
CR1, 2, 3, 4	Existing Counting Room Level 1 to 4 in PX24
C-RORC	Common Read-Out Received Card interfacing the DDL 2 to computer I/O bus
CRU	Common Read-out Unit
CTF	Data type: Compressed Time Frame
CTP	Central Trigger Processor
CWG	O^2 Computing Working Group

D

DAQ	Data Acquisition System
DCM	Data Collector Manager
DCS	Detector Control System
DDL1, 2	Detector Data Link successive releases
DDL SIU	DDL Source Interface Unit
DDP	Dynamic Disk Pool
DDR	Double Data Rate memory
DDS	Dynamic Deployment System
DIM	Distributed Information Management system
DM	Data Movers
DMA	Direct Memory Access
DNS	Domain Name Server
DRAM	Dynamic Random Access Memory
DSRO	Detector Specific Read-Out

E

EMC	Electromagnetic Calorimeter
EOS	CERN disk-based service providing a low latency storage infrastructure
EPN	Event Processing Node
ESD	Data type: Event Summary Data

F

FEE	Front-End Electronics
FDR	Infiniband Fourteen Data Rate
FIT	Fast Interaction Trigger
FLP	First Level Processor
FPGA	Field Programmable Gate Array
FSM	Finite-State Machine

G

GBT	GigaBit Transceiver
GEM	Gas Electron Multiplier
GPU	Graphics Processing Unit
GTU	Global Tracking Unit

H

HB	HeartBeat trigger
HBE	HeartBeat Event
HEP	High Energy Physics
HISTO	Data type: subset of AOD information specific for a given analysis
HLT	High-Level Trigger
HPC	High-Performance Computing
H-RORC	HLT Read-Out Received Card interfacing the DDL 1 to computer I/O bus
HTTP	Hyper Text Transfer Protocol

I

I/O	Input/Output
IB	Infiniband
IP	Internet Protocol
IPoIB	IP over IB
IP core	Intellectual Property core
iSCSI	Internet Small Computer System Interface
ITS	Inner Tracking System

J

JSON	JavaScript Object Notation

K

L

LHC	Large Hadron Collider
LoI	ALICE upgrade Letter of Intent
LS1, 2, 3	LHC Long Shutdown respectively in 2013-14, 2018-19, 2023-25
LTU	Local Trigger Unit

M

MC	Monte-Carlo
MC	Data type: simulated energy deposits in sensitive detectors
MCH	Muon Chamber System
MDB	Data format: Multiple Data Block
MDH	Data format: Multiple Data Header
MDT	Data format: Multiple Data Trailer
MID	Muon Identifier
MFT	Muon Forward Tracker
MTBF	Mean Time Between Failures

N

NAS	Network Attached Storage
NFS	A distributed Network File System and its associated network protocols
NUMA	Non-Uniform Memory Architecture

O

O^2	New online-offline computing system for the LS2 ALICE upgrade
OO	Object-Oriented
OSD	Object Storage Daemons
OSS	Object Storage Server in the Lustre file system

P

PC	Personal Computer
PCI	Peripheral Component Interconnect
PCI-X	Peripheral Component Interconnect eXtended
PCIe Gen1, 2, 3, 4	Peripheral Component Interconnect Express successive releases
PHS	Photon Spectrometer
PX24	Main access shaft to the eXperimental area at Point 2
PM25	Access shaft to the LHC Machine at Point 2

Q

QA	Quality Assurance
QC	Quality Control
QDR	Infiniband Quad Data Rate
QGP	Quark-Gluon Plasma
QSFP	Quad Small Form-factor Pluggable

R

RADOS	Ceph Reliable, Autonomic Distributed Object Store
RAID	Redundant Array of Inexpensive Disks or Redundant Array of Independent Disks
RAM	Random Access Memory
RMS	Resource Management System
RORC	Read-Out Received Card
Run 1, Run 2, Run 3	Periods of data taking operation of ALICE respectively in 2009-13, 2015-18, 2020-22.

S

SAN	Storage Area Network
SCADA	Supervisory Control and Data Acquisition system
SCD	TPC Space Charge Distortion
SDB	Data format: Single Data Block
SDH	Data format: Single Data Header
SDT	Data format: Single Data Trailer
SEPN	Super EPN
SMI	State Machine Interface
SoC	System on a Chip
SRAM	Static RAM
SSH	Secure Shell
S/N	Signal-to-Noise ratio
STF	Sub-Time Frame: a Time Frame containing data from a FLP

T

T1, T2, T3	Grid Tier 1, 2, 3
TDR	Technical Design Report
TF	Time Frame: the data from all data sources of a period of time
TOF	Time Of Flight detector
TQ	AliEn Task Queue
TPC	Time Projection Chamber
TRD	Transition Radiation Detector
TRG	Trigger
TTC	Timing Trigger and Control
TTS	Trigger and Timing distribution System

U

UDP	User Datagram Protocol
UPS	Uninterruptible Power Supply
UX25	Underground eXperimental area

V

VHDL	VHSIC Hardware Description Language
VHSIC	Very-High Speed Integrated Circuit

W

WLCG	Worlwide LHC Computing Grid

Z

ZDC	Zero Degree Calorimeter
ZeroMQ or 0MQ	High-performance asynchronous messaging library

Appendix B

Integrated luminosity requirements for proton–proton reference runs

The integrated luminosity required for the pp reference runs is estimated starting from the consideration that the statistical uncertainty on the pp reference has to be negligible with respect to that of the Pb–Pb measurement. In order to set numbers, the pp uncertainty is required to be $\sqrt{2}$ times smaller than the Pb–Pb uncertainty, so that the combined relative statistical uncertainty for e.g. a R_{AA} observable is at most 20% larger than the Pb–Pb uncertainty. Since the relative statistical uncertainty is the inverse of the statistical significance $1/\mathscr{S} = \sqrt{S+B}/S$, the requirement is:

$$\mathscr{S}_{\mathrm{pp}} = \sqrt{2} \cdot \mathscr{S}_{\mathrm{Pb-Pb}} \,. \tag{B.1}$$

This condition leads to different requirements in terms of statistics, depending on the S/B ratio, thus on the background level. For *high-background measurements* (which is the case for essentially all topics discussed in Chap. 2), $S \ll B$, thus $\mathscr{S} = S/\sqrt{B}$, and the condition becomes

$$S_{\mathrm{pp}}/\sqrt{B_{\mathrm{pp}}} = \sqrt{2} \cdot S_{\mathrm{Pb-Pb}}/\sqrt{B_{\mathrm{Pb-Pb}}}$$

thus

$$N_{\mathrm{pp}} = 2 \cdot N_{\mathrm{Pb-Pb}} \cdot [(\mathscr{S}/\sqrt{N})_{\mathrm{Pb-Pb}}/(\mathscr{S}/\sqrt{N})_{\mathrm{pp}}]^2$$

with N the number of events in pp and Pb–Pb (in a given centrality class, e.g. 0–20%).

In the following, the specific cases of several of the heavy flavour and quarkonium measurements are discussed.

Open heavy flavour

After the second long shut-down (LS2), the LHC is foreseen to accelerate the hadron beams at the nominal energy of $Z/A \cdot 7$ TeV, and the centre-of-mass energy will be $\sqrt{s} = 14$ TeV for pp and $\sqrt{s_{\mathrm{NN}}} = 5.5$ TeV for Pb–Pb collisions. If the pp reference data are collected at 14 TeV, the charm and beauty production cross sections have to be scaled to the Pb–Pb energy, in order to define the nuclear modification factors $R_{\mathrm{AA}}(p_{\mathrm{T}})$. The scaling factors can be obtained using perturbative QCD calculations, like FONLL [1]. The definition of the scaling factor and of its theoretical uncertainty are described in [2]. Figure B.1 shows the relative uncertainties of the scaling factors for D mesons (left) and B mesons (right) from 14 to 5.5 TeV. The scaling factors and their uncertainties for Λ_{c} and Λ_{b} baryons are the same as for D and B mesons, respectively. For the case of charm at low p_{T}, the uncertainty is larger than 50% ($p_{\mathrm{T}} < 2$ GeV/c). This would be the dominant uncertainty in the measurement of the D and Λ_{c} nuclear modification factors. Therefore, for charm, the reference data should be collected with pp collisions at $\sqrt{s} = 5.5$ TeV. For

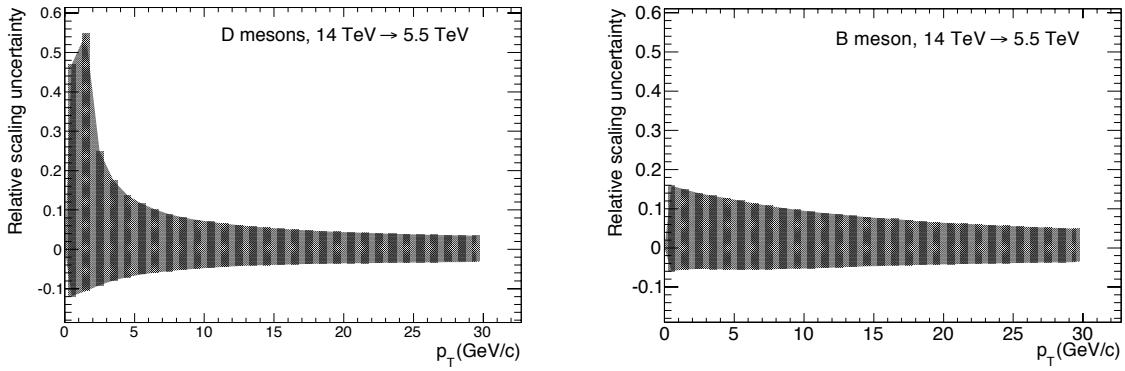

Figure B.1: Relative uncertainty for the scaling factor of the cross section of D (left) and B (right) mesons from $\sqrt{s} = 14$ to 5.5 TeV using FONLL [1].

beauty hadrons, the uncertainty of the scaling factor is smaller than 10–15%. Therefore, it is not expected to be larger than the systematic uncertainties of the measurements in Pb–Pb and pp collisions. This gives the possibility to use, for the reference measurement of B^+ and Λ_b, proton–proton collisions at full LHC energy, where the beauty production cross section is larger by a factor two to three, thus a correspondingly lower integrated luminosity is required.

The required integrated luminosity in pp collisions was estimated using the *high-background measurements* procedure described at the beginning of this section, for the following measurements: D^0, Λ_c and beauty production measurement via non-prompt J/ψ (the latter is described in the next section together with the charmonium measurements). Estimates are in progress for the other beauty hadron studies described in [3], namely B^+ and Λ_b.

The estimation for the D^0 meson is described in ITS Upgrade CDR and it corresponds to $L_{\rm int} = 6$ pb^{-1} at $\sqrt{s} = 5.5$ TeV [4] .

For Λ_c the following procedure was used. The significance per event in central Pb–Pb collisions, $(\mathscr{S}/\sqrt{N})_{\rm Pb-Pb}$, was taken from the ITS Upgrade TDR [3]. The significance per event in pp collisions, $(\mathscr{S}/\sqrt{N})_{\rm pp}$, was estimated by multiplying $(\mathscr{S}/\sqrt{N})_{\rm Pb-Pb}$ with a correction factor for the signal per event and a correction factor for the background per event. The former is the inverse of the product of the average number of binary nucleon–nucleon collisions in Pb–Pb central collisions and the ($p_{\rm T}$-dependent) nuclear modification factor $R_{\rm AA}$ that was assumed for the Pb–Pb studies of each particle. The latter, the scaling factor for the background, was estimated as the ($p_{\rm T}$-dependent) ratio of the background yield in pp simulations and in Pb–Pb simulations at the same energy. The resulting integrated luminosity that fulfils the requirement of $\sqrt{2}$-larger significance in pp than in Pb–Pb depends slightly on $p_{\rm T}$. The maximum value was defined as the required integrated luminosity for the measurement. The resulting value is $L_{\rm int} = 5$ pb^{-1} at $\sqrt{s} = 5.5$ TeV. Table B.1 reports the projected statistical uncertainties for Pb–Pb collisions (10 nb^{-1}) and pp collisions.

Table B.1: Statistical uncertainties for Λ_c production measurement in central Pb–Pb collisions (10 nb^{-1}) [3] and pp collisions, for a subset of the $p_{\rm T}$ intervals covered by the measurement.

$p_{\rm T}$	Λ_c	
(GeV/c)	Pb–Pb 0–20%	pp, 5.5 TeV, 5 pb^{-1}
2–4	12%	8%
6–8	4%	2%
7–10	4%	3%

Charmonium

For charmonium measurements the proton–proton reference should be collected at the same centre-of-mass energy as the Pb–Pb data, because the systematic uncertainty of the energy scaling from 14 to

5.5 TeV would be similar as for open charm measurements, namely up to 30–50% for p_{T} close 0.

For charmonium measurements in the dielectron decay channel at central rapidity the signal-to-background ratio is small. For example, for J/ψ in central Pb–Pb collisions, it ranges from 0.01 at $p_{\mathrm{T}} < 1$ GeV$/c$ to 0.20 at 10 GeV$/c$. Therefore, the relation for *high-background measurements* discussed at the beginning of this section was used. The significance per event in pp collisions, $(\mathscr{S}/\sqrt{N})_{\mathrm{pp}}$, was taken from the analysis of inclusive and non-prompt J/ψ production from pp data at $\sqrt{s} = 7$ TeV [5]. It was assumed that the significance per event is the same at $\sqrt{s} = 7$ and 5.5 TeV and that it is the same with the present ALICE detector layout and with the upgraded detector. The latter is a realistic assumption for inclusive J/ψ and a conservative one for non-prompt J/ψ from B meson decays, for which the significance per event is expected to be larger with the improved tracking precision of the new ITS detector. The resulting integrated luminosity required for pp collisions at $\sqrt{s} = 5.5$ TeV is of about 5 pb^{-1}.

For charmonium measurements at forward rapidity with the MFT and the MUON spectrometer, the inclusive J/ψ and $\psi(2\mathrm{S})$ significance per event for pp collisions at $\sqrt{s} = 5.5$ TeV was conservatively taken to be the same as for measurements with the MUON spectrometer alone at $\sqrt{s} = 7$ TeV [6]. The stastistical uncertainty for non-prompt J/ψ was assumed to scaled from Pb–Pb to pp by the same factor as the one for inclusive J/ψ.

The statistical uncertainties for Pb–Pb collisions (10 nb^{-1}) and pp collisions (5 pb^{-1}) are reported in Tab. B.2.

Table B.2: Statistical uncertainties for inclusive and non-prompt J/ψ and for inclusive $\psi(2\mathrm{S})$, at central and forward rapidity, in Pb–Pb collisions (10 nb^{-1}) and pp collisions (5 pb^{-1}) at $\sqrt{s_{\mathrm{NN}}} = 5.5$ TeV, for a subset of the p_{T} intervals covered by the measurements. The text 'not av.' indicates that the estimate of the uncertainty is not available, the text 'not acc.' indicates that the measurement is not accessible for a given p_{T} interval. The statistical uncertainties for Pb–Pb collisions are extracted from [7] for inclusive J/ψ and $\psi(2\mathrm{S})$ at central rapidity, from [8] for inclusive J/ψ and $\psi(2\mathrm{S})$ at forward rapidity, from [3] for non-prompt J/ψ at central rapidity, and from [9] for non-prompt J/ψ at forward rapidity. The statistical uncertainties for pp collisions are estimated starting from those in [5] and [6] for central and forward rapidity, respectively (more details are given in the text).

p_{T}	Incl. J/ψ		Non-prompt J/ψ		Incl. $\psi(2\mathrm{S})$	
(GeV$/c$)	Pb–Pb 0–10%	pp	Pb–Pb 0–10%	pp	Pb–Pb 0–10%	pp
Central rapidity, dielectron channel						
> 0	not av.	not av.	not av.	not av.	7%	5%
0–1	1%	0.5%	not acc.	not acc.	not av.	not av.
1–2	0.5%	0.5%	5%	3.5%	not av.	not av.
3–4	1%	0.7%	4%	1.5%	not av.	not av.
7–8	3%	2%	6%	3%	not av.	not av.
Forward rapidity, dimuon channel						
> 0	0.07%	0.30%	not av.	not av.	4.3%	3.0%
0–1	0.23%	0.75%	1.5%	1.3%	10.7%	7.5%
4–5	0.18%	0.75%	not av.	not av.	10.9%	7.5%

Appendix C

Tools and procedures

Two working groups (CWG2 on Tools, Procedures and Guidelines and CWG11 on Software Lifecycle) have been established in order to respectively evaluate common tools for the O^2 project and tools belonging to the software lifecycle.

C.1 Evaluation procedure

In order to set the basis for the evaluation of tools, the CWG2 has set a procedure to be followed when the need for a tool is identified and a selection shoud be done. The evaluation procedure has been proposed to all working groups, approved and can be summarized as follows:

- The person or group (i.e. "evaluator") who carries out the evaluation should contact the CWG2 to know whether the tool is already being surveyed by someone.
- The evaluator should define the problem and identify a list of requirements.
- The evaluator should gather specifications by getting input from at least the three projects of the O^2 framework (HLT, DAQ, Offline). The specs gathering is an opportunity to get experience from them.
- The evaluator should prepare a list of candidate tools. The list of requirements should be sent to the CWG2 for information, as well as the list of tools and the possible candidate for a shortcut.
- The CWG2 will quickly check that the procedure has been followed and that the 3 projects have been contacted for the requirements gathering.
- If one tool that covers the requirements is identified, the proposal can be presented to the O^2 plenary meeting after the CWG2 has been informed.
- If a consensus is found at the meeting, the proposal is accepted as official, an evaluation will follow otherwise.
- If there are several candidate tools in the shortlist, the evaluator will proceed with identifying specific criteria that will be used to compare the shortlisted tools. He/she will start prototyping (if needed) and evaluate the tools in the shortlist.
- The evaluator will share his/her proposal with all the working groups members via email and, subsequently, present the findings at the O^2 plenary meeting.
- Eventually, an implementation of the proposed solution will follow.

Figure C.1 shows the schematic of the evaluation procedure.

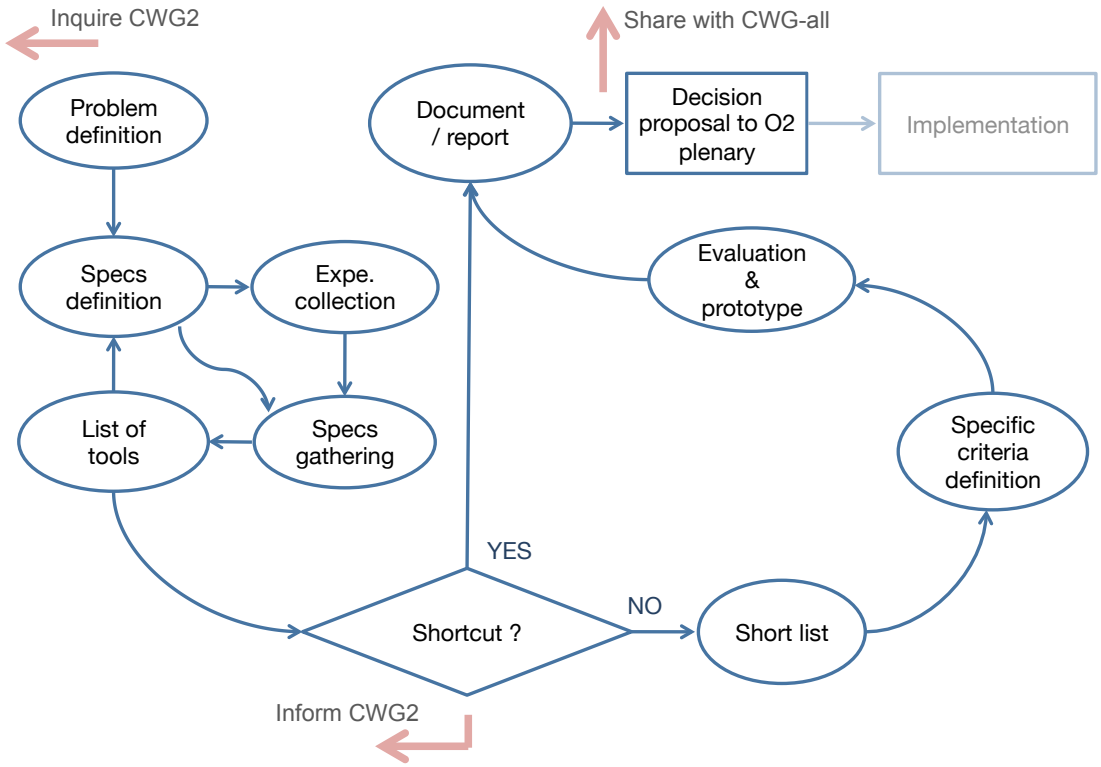

Figure C.1: The evaluation procedure

C.2 Selected tools

While establishing the evaluation procedure, the CWG2 has identified a list of tools that needed to be selected in order to carry out activities, which would be common to other working groups. The tools in the list have been then categorized by priority. The next sections describe the evaluation of different parameters that has brought to the selection of tools covering the following tasks:

- Support: Issue/Bug tracking and version control.
- Documentation: Website creation and source code documentation.
- Software: build and computing simulation.

C.2.1 Issue/Bug tracking system: JIRA

An evaluation of different tools for bug and issue tracking systems have been carried out.

In particular, the bug tracking part can be considered as a specialization of the issue tracking system since it is focused on software bugs, improvements and tasks. The issue tracking side should manage activities, issues and interventions, mostly related to the experiment's operations and system administration activities. Specifications have been collected and resulted in a list of mandatory requirements that are summarized below.

- Users:

  - Read access to the ALICE collaborations (about 1500 users)
  - Write access to the ALICE O^2 project developers and system administrators (about 100 users)
  - Incremental addition of users
- Authentication/authorization: Users shall be able to use their CERN NICE credentials. It should be possible to use the CERN e-groups as well. It implies a way to define groups and/or roles.
- Multiple projects: the tool should allow creating multiple projects.
- Software versioning: It should be possible to associate versions to projects.
- Sub-projects / components: the issue tracking system should be able to reflect a structure divided into several modules.
- API: the tools should provide an interface (API) for integration with other systems for automatic issues creation and resolving, or extraction of information
- Integration with the version control system: we expect the tool to be able to show commits related to a ticket (consider at least SVN [2**] and Git [3**] integration)
- Customization: The issue workflow should be customizable according to issue type and/or project. It means that the issues states, and their transitions, can be modified to fit specific use cases.
- Prioritization of issues
- Notification capabilities

Several tools have been considered as candidates for bug/issue tracking and JIRA [1] resulted meeting the requirements.

JIRA has been proposed to the computing working groups and accepted.

C.2.2 Version control system: Git

An evaluation was carried in order to find a tool able to track and provide control over changes of the project source code and to facilitate the collaboration between ALICE O^2 software developers. The evaluation was based on the specifications and requirements collected by CWG2 from ALICE O^2 collaboration:

- Integration with CERN Central Services
  - Authentication and authorization using CERN credentials
  - User management with CERN egroups
  - 24/7 support (service hosting, support, daily backups)
  - Service web interface synchronized with CERN authentication and authorization
- Ability to store large repositories (more than 1GB of sources)
- Multiple operating systems support with mandatory requirements for CERN SLCxxx
- Atomic commits - a set of distinct changes is applied as a single operation
- Capability to handle locks
- Renamed/copied/moved/removed files/directory retain full revision history

- Native support for binary files
- Hooks - the possibility to run different scripts before or after an operation on the repository
- Possibility to export automatically sources as an archive or different format
- Dated checkouts (get repository trunk snapshot at given date/time - mostly necessary for auto build system, to make things reproducible)
- Purge capabilities
- Capability to import from former VCS into the new one

CWG2 followed the *Evaluation Procedure* and found **Git** [2] the most appropriate tool to handle the Version Control System for ALICE O^2 collaboration.

C.2.3 Website creation tool: Drupal

Tools for creating, editing and publishing websites have also been evaluated. The focus was put on how to make the websites content available online to the ALICE O^2 community, how to easily add and maintain online content and how to integrate content generated by external tools. Specifications have been collected and resulted in a list of mandatory requirements there are summarized below.

- Easy Editing: the tool must allow users to create or edit content in a simple and easy way. Even users without knowledge of HTML should be able to contribute to the website, ideally via a WYSIWYG editor. It must be nevertheless possible to add directly HTML and/or specific markup languages.
- Content Organization: the tool must allow for an efficient content structuring and the development of an intuitive website hierarchy and navigation.
- Efficient Search: the tool must provide an efficient search engine that allows users to find pages without having to navigate the website.
- Revision Control: the tool must allow users to access the change history of a page, see the differences between versions of a page and watch pages for changes, ideally with email notification of changes.
- Look and Feel: the tool must provide an attractive Look and Feel by default, ideally via selectable themes.
- Access Control: the tool must provide an Access Control mechanism, allowing the following cases: no access, read-only access, write access per page or group of pages.
- Integration with CERN Authentication services: users should be able to use their CERN credentials to authenticate.
- Integration with external tools: the tool must provide a mechanism to automatically import content generated by external tools, in particular the tool which will be selected for Source Code documentation.
- Print: the tool must be able to generate a printer-friendly version of a page.

Based on these requirements and on the collected experience in ALICE, the other LHC experiments and CERN IT, the selected tool for website creation was Drupal [3].

C.2.4 Source code documentation tool: Doxygen

An evaluation on source code documentation tools has been carried out. The evaluation of the Source code documentation tool was based on the specifications and requirements collected by CWG2 from ALICE O^2 collaboration:

- Documentation embedded in source code: the tool must be able to extract and generate the documentation from source code files comments and code.
- Online output format: the tool must be able to generate documentation in a web-based format such as HTML.
- Offline output format: the tool must be able to generate documentation in an offline format such as PDF or LaTeX.
- Attractive output: the tool must generate documentation with an attractive look and feel.
- Flexibility in formatting, rich syntax: the tool must allow flexibility in the documentation formatting, including the usage of rich text markup.
- Simplicity: minimal amount of macros/hooks needed when documenting the code.

Based on these requirements and on the collected experience in ALICE, the other LHC experiments and ROOT and Geant4 projects, the selected tool for source code documentation was Doxygen [4].

C.2.5 Software build system: Cmake

- C/C++ Software build system
  Following the *Evaluation Procedure*, CWG11 tried to find a C/C++ build system that would:
  - control C/C++ software compilation, the process of converting source code into binaries
  - install resulting binaries into a convenient format

  The evaluation was based on the requirements and specifications received from the ALICE O^2 collaboration.
  - Incremental builds, the system should be able to build only the modified code and its dependencies
  - Parallel building support for multi core systems
  - Support for source code dependency management, the system should be able to define dependencies between the code and external libraries but also inside the source tree
  - Verbose error reporting and tracking
  - Cross-compilation, the system should be capable of creating executable code for a platform other than the one on which the compiler is running
  - Several builds from the same source tree using different build configurations
  - Run on different platforms such as Linux, Mac OS and Window
  - Configurability, the possibility to enable or disable different features or compilation flags
  - Easy to use and implement
  - Taking in consideration the size of the project, build speed is an important criteria
  - Generate build configuration for different IDEs, such as XCode or Visual Studio
  - Built-in possibility to create binary distributions (RPM, deb, etc.)

  The chosen C/C++ software build system was **CMake** [5].

C.2.6 Computing simulation tool: Omnet++ and the Monarc simulation tool

The choice was made to use a discrete-event simulation tool, which models the operation of a system as a discrete sequence of events in time because it fits perfectly the needs. There is a large number of tools available to create simulations of computing system. The evaluation of the tools was based on the evaluation of the following criterias:

- Scalability: it should be able to simulate efficiently of the order 2000 nodes.
- Ease of use: to facilitate the creation of the simulations.
- Price of a licence.
- Availability on Linux.
- Quality of the GUI.
- Possibility to create plots.

In order to select the best possible tool for our purpose, a survey has been done. More than 20 tools where considered, and 5 thoroughly tested: FlexSim (trial version), Omnet++, Ptolemy II, Simpy, and SystemC. Although a bit less efficient than SystemC in terms of speed, the ease of use, the possibility to reuse very easily network modules (TCP/IP, ethernet, etc.), and the possibility to display data plots nicely, lead us to choose Omnet++.

In addition to Omnet++, a tool developed by the Monarc project for the Grid simulation, is also used for some high-level simulations. The simulation and modeling task for O^2 system requires to describe complex data flow patterns, running on large scale distributed systems and exchanging very large amounts of data. A process oriented approach [6] for discrete event simulation is well suited to model precisely large number of concurrent I/O programs as well as all the stochastic patterns in the DAQ systems. Threaded objects or Active Objects (having an execution thread, programme counter, stack, mutual exclusion mechanism...) offer much great flexibility in simulating concurrent, large scale distributed system in an effective way.

C.3 Policies

C.3.1 C++ Coding conventions

CWG2 proposed a series of coding conventions for the C++ language covering two main aspects:

- Coding guidelines.
- Coding style.

The goal of the coding guidelines is to provide dos and don'ts of writing C++ code that allow the O^2 programmer to manage the complexity of the C++ language. The C++ coding guidelines cover different aspects of the C++ language such as Header files, namespaces, scoping, classes, etc.

The goal of the coding style is to provide a number of rules that keep the code base manageable by enforcing consistency. Constistency across the O^2 code allows any programmer to look at another programmer's code and understand it quickly. Moreover, properly documenting the code ensures the code's readability and helps efficiency. The coding style is divided in three different parts:

- A section on "Naming" that consists of general guidelines on how to name variables, files, functions, enumerators, macros, etc.

- A section on ”Formatting”, which includes rules on line length, spaces, braces and other aspects.
- A section on ”Comments” that covers what to document and how.

Appendix D

The ALICE Collaboration

J. Adam[40] , D. Adamová[83] , M.M. Aggarwal[87] , G. Aglieri Rinella[36] , M. Agnello[111] , N. Agrawal[48] , Z. Ahammed[132] , S.U. Ahn[68] , I. Aimo[94 ,111] , S. Aiola[137] , M. Ajaz[16] , A. Akindinov[58] , S.N. Alam[132] , D. Aleksandrov[100] , B. Alessandro[111] , D. Alexandre[102] , R. Alfaro Molina[64] , A. Alici[105 ,12] , A. Alkin[3] , J. Alme[38] , T. Alt[43] , S. Altinpinar[18] , I. Altsybeev[131] , C. Alves Garcia Prado[120] , C. Andrei[78] , A. Andronic[97] , V. Anguelov[93] , J. Anielski[54] , T. Antičić[98] , F. Antinori[108] , P. Antonioli[105] , L. Aphecetche[113] , H. Appelshäuser[53] , S. Arcelli[28] , N. Armesto[17] , R. Arnaldi[111] , T. Aronsson[137] , I.C. Arsene[22] , M. Arslandok[53] , A. Augustinus[36] , R. Averbeck[97] , M.D. Azmi[19] , M. Bach[43] , A. Badalà[107] , Y.W. Baek[44] , S. Bagnasco[111] , R. Bailhache[53] , R. Bala[90] , A. Baldisseri[15] , F. Baltasar Dos Santos Pedrosa[36] , R.C. Baral[61] , A.M. Barbano[111] , R. Barbera[29] , F. Barile[33] , G.G. Barnaföldi[136] , L.S. Barnby[102] , V. Barret[70] , P. Bartalini[7] , K. Barth[36] , J. Bartke[117] , E. Bartsch[53] , M. Basile[28] , N. Bastid[70] , S. Basu[132] , B. Bathen[54] , G. Batigne[113] , A. Batista Camejo[70] , B. Batyunya[66] , P.C. Batzing[22] , I.G. Bearden[80] , H. Beck[53] , C. Bedda[111] , N.K. Behera[49 ,48] , I. Belikov[55] , F. Bellini[28] , H. Bello Martinez[2] , R. Bellwied[122] , R. Belmont[135] , E. Belmont-Moreno[64] , V. Belyaev[76] , G. Bencedi[136] , S. Beole[27] , I. Berceanu[78] , A. Bercuci[78] , Y. Berdnikov[85] , D. Berenyi[136] , R.A. Bertens[57] , D. Berzano[36 ,27] , L. Betev[36] , A. Bhasin[90] , I.R. Bhat[90] , A.K. Bhati[87] , B. Bhattacharjee[45] , J. Bhom[128] , L. Bianchi[122] , N. Bianchi[72] , C. Bianchin[135 ,57] , J. Bielčík[40] , J. Bielčíková[83] , A. Bilandzic[80] , R. Biswas[4] , S. Biswas[79] , S. Bjelogrlic[57] , F. Blanco[10] , D. Blau[100] , C. Blume[53] , F. Bock[74 ,93] , A. Bogdanov[76] , H. Bøggild[80] , L. Boldizsár[136] , M. Bombara[41] , J. Book[53] , H. Borel[15] , A. Borissov[96] , M. Borri[82] , F. Bossú[65] , M. Botje[81] , E. Botta[27] , S. Böttger[52] , P. Braun-Munzinger[97] , M. Bregant[120] , T. Breitner[52] , T.A. Broker[53] , T.A. Browning[95] , M. Broz[40] , E.J. Brucken[46] , E. Bruna[111] , G.E. Bruno[33] , D. Budnikov[99] , H. Buesching[53] , S. Bufalino[111 ,36] , P. Buncic[36] , O. Busch[93 ,128] , Z. Buthelezi[65] , J.T. Buxton[20] , D. Caffarri[36] , X. Cai[7] , H. Caines[137] , L. Calero Diaz[72] , A. Caliva[57] , E. Calvo Villar[103] , P. Camerini[26] , F. Carena[36] , W. Carena[36] , J. Castillo Castellanos[15] , A.J. Castro[125] , E.A.R. Casula[25] , C. Cavicchioli[36] , C. Ceballos Sanchez[9] , J. Cepila[40] , P. Cerello[111] , J. Cerkala[115] , B. Chang[123] , S. Chapeland[36] , M. Chartier[124] , J.L. Charvet[15] , S. Chattopadhyay[132] , S. Chattopadhyay[101] , V. Chelnokov[3] , M. Cherney[86] , C. Cheshkov[130] , B. Cheynis[130] , V. Chibante Barroso[36] , D.D. Chinellato[121] , P. Chochula[36] , K. Choi[96] , M. Chojnacki[80] , S. Choudhury[132] , P. Christakoglou[81] , C.H. Christensen[80] , P. Christiansen[34] , T. Chujo[128] , S.U. Chung[96] , Z. Chunhui[57] , C. Cicalo[106] , L. Cifarelli[12 ,28] , F. Cindolo[105] , J. Cleymans[89] , F. Colamaria[33] , D. Colella[33] , A. Collu[25] , M. Colocci[28] , G. Conesa Balbastre[71] , Z. Conesa del Valle[51] , M.E. Connors[137] , J.G. Contreras[11 ,40] , T.M. Cormier[84] , Y. Corrales Morales[27] , I. Cortés Maldonado[2] , P. Cortese[32] , M.R. Cosentino[120] , F. Costa[36] , P. Crochet[70] , R. Cruz Albino[11] , E. Cuautle[63] , L. Cunqueiro[36] , T. Dahms[92 ,37] , A. Dainese[108] , A. Danu[62] , D. Das[101] , I. Das[51 ,101] , S. Das[4] , A. Dash[121] , S. Dash[48] , S. De[120] , A. De Caro[31 ,12] , G. de Cataldo[104] , J. de Cuveland[43] , A. De Falco[25] , D. De Gruttola[12 ,31] , N. De Marco[111] , S. De Pasquale[31] , A. Deisting[97 ,93] , A. Deloff[77] , E. Dénes[136] , G. D'Erasmo[33] , D. Di Bari[33] , A. Di Mauro[36] , P. Di Nezza[72] , M.A. Diaz Corchero[10] , T. Dietel[89] , P. Dillenseger[53] , R. Divià[36] , Ø. Djuvsland[18] , A. Dobrin[57 ,81] , T. Dobrowolski[77 ,i] , D. Domenicis Gimenez[120] , B. Dönigus[53] , O. Dordic[22] , A.K. Dubey[132] , A. Dubla[57] , L. Ducroux[130] , P. Dupieux[70] , R.J. Ehlers[137] , D. Elia[104] , H. Engel[52] , B. Erazmus[36 ,113] , I. Erdemir[53] , F. Erhardt[129] , D. Eschweiler[43] , B. Espagnon[51] , M. Estienne[113] , S. Esumi[128] , J. Eum[96] , D. Evans[102] , S. Evdokimov[112] , G. Eyyubova[40] , L. Fabbietti[37 ,92] , D. Fabris[108] , J. Faivre[71] , A. Fantoni[72] , M. Fasel[74] , L. Feldkamp[54] , D. Felea[62] , A. Feliciello[111] , G. Feofilov[131] , J. Ferencei[83] , A. Fernández Téllez[2] , E.G. Ferreiro[17] , A. Ferretti[27] , A. Festanti[30] , J. Figiel[117] , M.A.S. Figueredo[124] , S. Filchagin[99] , D. Finogeev[56] , F.M. Fionda[104] , E.M. Fiore[33] , M.G. Fleck[93] , M. Floris[36] ,

S. Foertsch[65] , P. Foka[97] , S. Fokin[100] , E. Fragiacomo[110] , A. Francescon[30 ,36] , U. Frankenfeld[97] , U. Fuchs[36] , C. Furget[71] , A. Furs[56] , M. Fusco Girard[31] , J.J. Gaardhøje[80] , M. Gagliardi[27] , A.M. Gago[103] , M. Gallio[27] , D.R. Gangadharan[74] , P. Ganoti[88] , C. Gao[7] , C. Garabatos[97] , E. Garcia-Solis[13] , C. Gargiulo[36] , P. Gasik[92 ,37] , M. Germain[113] , A. Gheata[36] , M. Gheata[62 ,36] , P. Ghosh[132] , S.K. Ghosh[4] , P. Gianotti[72] , P. Giubellino[36] , P. Giubilato[30] , E. Gladysz-Dziadus[117] , P. Glässel[93] , A. Gomez Ramirez[52] , P. González-Zamora[10] , S. Gorbunov[43] , L. Görlich[117] , S. Gotovac[116] , V. Grabski[64] , L.K. Graczykowski[134] , K.L. Graham[102] , A. Grelli[57] , A. Grigoras[36] , C. Grigoras[36] , V. Grigoriev[76] , A. Grigoryan[1] , S. Grigoryan[66] , B. Grinyov[3] , N. Grion[110] , J.F. Grosse-Oetringhaus[36] , J.-Y. Grossiord[130] , R. Grosso[36] , F. Guber[56] , R. Guernane[71] , B. Guerzoni[28] , K. Gulbrandsen[80] , H. Gulkanyan[1] , T. Gunji[127] , A. Gupta[90] , R. Gupta[90] , R. Haake[54] , Ø. Haaland[18] , C. Hadjidakis[51] , M. Haiduc[62] , H. Hamagaki[127] , G. Hamar[136] , L.D. Hanratty[102] , A. Hansen[80] , J.W. Harris[137] , H. Hartmann[43] , A. Harton[13] , D. Hatzifotiadou[105] , S. Hayashi[127] , S.T. Heckel[53] , M. Heide[54] , H. Helstrup[38] , A. Herghelegiu[78] , G. Herrera Corral[11] , B.A. Hess[35] , K.F. Hetland[38] , T.E. Hilden[46] , H. Hillemanns[36] , B. Hippolyte[55] , P. Hristov[36] , M. Huang[18] , T.J. Humanic[20] , N. Hussain[45] , T. Hussain[19] , D. Hutter[43] , D.S. Hwang[21] , R. Ilkaev[99] , I. Ilkiv[77] , M. Inaba[128] , C. Ionita[36] , M. Ippolitov[76 ,100] , M. Irfan[19] , M. Ivanov[97] , V. Ivanov[85] , V. Izucheev[112] , P.M. Jacobs[74] , S. Jadlovska[115] , C. Jahnke[120] , H.J. Jang[68] , M.A. Janik[134] , P.H.S.Y. Jayarathna[122] , C. Jena[30] , S. Jena[122] , R.T. Jimenez Bustamante[97] , P.G. Jones[102] , H. Jung[44] , A. Jusko[102] , P. Kalinak[59] , A. Kalweit[36] , J. Kamin[53] , J.H. Kang[138] , V. Kaplin[76] , S. Kar[132] , A. Karasu Uysal[69] , O. Karavichev[56] , T. Karavicheva[56] , E. Karpechev[56] , U. Kebschull[52] , R. Keidel[139] , D.L.D. Keijdener[57] , M. Keil[36] , K.H. Khan[16] , M.M. Khan[19] , P. Khan[101] , S.A. Khan[132] , A. Khanzadeev[85] , Y. Kharlov[112] , B. Kileng[38] , B. Kim[138] , D.W. Kim[44 ,68] , D.J. Kim[123] , H. Kim[138] , J.S. Kim[44] , M. Kim[44] , M. Kim[138] , S. Kim[21] , T. Kim[138] , S. Kirsch[43] , I. Kisel[43] , S. Kiselev[58] , A. Kisiel[134] , G. Kiss[136] , J.L. Klay[6] , C. Klein[53] , J. Klein[93] , C. Klein-Bösing[54] , A. Kluge[36] , M.L. Knichel[93] , A.G. Knospe[118] , T. Kobayashi[128] , C. Kobdaj[114] , M. Kofarago[36] , T. Kollegger[97 ,43] , A. Kolojvari[131] , V. Kondratiev[131] , N. Kondratyeva[76] , E. Kondratyuk[112] , A. Konevskikh[56] , M. Kopcik[115] , C. Kouzinopoulos[36] , O. Kovalenko[77] , V. Kovalenko[131] , M. Kowalski[117] , S. Kox[71] , G. Koyithatta Meethaleveedu[48] , J. Kral[123] , I. Králik[59] , A. Kravčáková[41] , M. Krelina[40] , M. Kretz[43] , M. Krivda[102 ,59] , F. Krizek[83] , E. Kryshen[36] , M. Krzewicki[97 ,43] , A.M. Kubera[20] , V. Kučera[83] , T. Kugathasan[36] , C. Kuhn[55] , P.G. Kuijer[81] , I. Kulakov[43] , J. Kumar[48] , L. Kumar[79 ,87] , P. Kurashvili[77] , A. Kurepin[56] , A.B. Kurepin[56] , A. Kuryakin[99] , S. Kushpil[83] , M.J. Kweon[50] , Y. Kwon[138] , S.L. La Pointe[111] , P. La Rocca[29] , C. Lagana Fernandes[120] , I. Lakomov[51 ,36] , R. Langoy[42] , C. Lara[52] , A. Lardeux[15] , A. Lattuca[27] , E. Laudi[36] , R. Lea[26] , L. Leardini[93] , G.R. Lee[102] , S. Lee[138] , I. Legrand[36] , R.C. Lemmon[82] , V. Lenti[104] , E. Leogrande[57] , I. León Monzón[119] , M. Leoncino[27] , P. Lévai[136] , S. Li[7 ,70] , X. Li[14] , J. Lien[42] , R. Lietava[102] , S. Lindal[22] , V. Lindenstruth[43] , C. Lippmann[97] , M.A. Lisa[20] , H.M. Ljunggren[34] , D.F. Lodato[57] , P.I. Loenne[18] , V.R. Loggins[135] , V. Loginov[76] , C. Loizides[74] , X. Lopez[70] , E. López Torres[9] , A. Lowe[136] , P. Luettig[53] , M. Lunardon[30] , G. Luparello[26] , P.H.F.N.D. Luz[120] , A. Maevskaya[56] , M. Mager[36] , S. Mahajan[90] , S.M. Mahmood[22] , A. Maire[55] , R.D. Majka[137] , M. Malaev[85] , I. Maldonado Cervantes[63] , L. Malinina[66] , D. Mal'Kevich[58] , P. Malzacher[97] , A. Mamonov[99] , L. Manceau[111] , V. Manko[100] , F. Manso[70] , V. Manzari[104 ,36] , M. Marchisone[27] , J. Mareš[60] , G.V. Margagliotti[26] , A. Margotti[105] , J. Margutti[57] , A. Marín[97] , C. Markert[118] , M. Marquard[53] , N.A. Martin[97] , J. Martin Blanco[113] , P. Martinengo[36] , M.I. Martínez[2] , G. Martínez García[113] , M. Martinez Pedreira[36] , Y. Martynov[3] , A. Mas[120] , S. Masciocchi[97] , M. Masera[27] , A. Masoni[106] , L. Massacrier[113] , A. Mastroserio[33] , H. Masui[128] , A. Matyja[117] , C. Mayer[117] , J. Mazer[125] , M.A. Mazzoni[109] , D. Mcdonald[122] , F. Meddi[24] , A. Menchaca-Rocha[64] , E. Meninno[31] , J. Mercado Pérez[93] , M. Meres[39] , Y. Miake[128] , M.M. Mieskolainen[46] , K. Mikhaylov[58 ,66] , L. Milano[36] , J. Milosevic[22 ,133] , L.M. Minervini[104 ,23] , A. Mischke[57] , A.N. Mishra[49] , D. Miśkowiec[97] , J. Mitra[132] , C.M. Mitu[62] , N. Mohammadi[57] , B. Mohanty[79 ,132] , L. Molnar[55] , L. Montaño Zetina[11] , E. Montes[10] , M. Morando[30] , D.A. Moreira De Godoy[113] , S. Moretto[30] , A. Morreale[113] , A. Morsch[36] , V. Muccifora[72] , E. Mudnic[116] , D. Mühlheim[54] , S. Muhuri[132] , M. Mukherjee[132] , H. Müller[36] , J.D. Mulligan[137] , M.G. Munhoz[120] , S. Murray[65] , L. Musa[36] , J. Musinsky[59] , B.K. Nandi[48] , R. Nania[105] , E. Nappi[104] , M.U. Naru[16] , C. Nattrass[125] , K. Nayak[79] , T.K. Nayak[132] , S. Nazarenko[99] , A. Nedosekin[58] , L. Nellen[63] , F. Ng[122] , M. Nicassio[97] , M. Niculescu[62 ,36] , J. Niedziela[36] , B.S. Nielsen[80] , S. Nikolaev[100] , S. Nikulin[100] , V. Nikulin[85] , F. Noferini[105 ,12] , P. Nomokonov[66] , G. Nooren[57] , J. Norman[124] , A. Nyanin[100] , J. Nystrand[18] , H. Oeschler[93] , S. Oh[137] , S.K. Oh[67] , A. Ohlson[36] , A. Okatan[69] , T. Okubo[47] , L. Olah[136] , J. Oleniacz[134] , A.C. Oliveira Da Silva[120] , M.H. Oliver[137] , J. Onderwaater[97] , C. Oppedisano[111] , A. Ortiz Velasquez[63] , A. Oskarsson[34] , J. Otwinowski[97 ,117] , K. Oyama[93] , M. Ozdemir[53] , Y. Pachmayer[93] , P. Pagano[31] , G. Paić[63] , C. Pajares[17] , S.K. Pal[132] , J. Pan[135] , A.K. Pandey[48] , D. Pant[48] , P. Papcun[115] , V. Papikyan[1] , G.S. Pappalardo[107] , P. Pareek[49] , W.J. Park[97] , S. Parmar[87] , A. Passfeld[54] , V. Paticchio[104] , R.N. Patra[132] , B. Paul[101] , T. Peitzmann[57] , H. Pereira Da Costa[15] , E. Pereira De Oliveira Filho[120] , D. Peresunko[76 ,100] ,

C.E. Pérez Lara[81] , V. Peskov[53] , Y. Pestov[5] , V. Petráček[40] , V. Petrov[112] , M. Petrovici[78] , C. Petta[29] , S. Piano[110] , M. Pikna[39] , P. Pillot[113] , O. Pinazza[105 ,36] , L. Pinsky[122] , D.B. Piyarathna[122] , M. Płoskoń[74] , M. Planinic[129] , J. Pluta[134] , S. Pochybova[136] , P.L.M. Podesta-Lerma[119] , M.G. Poghosyan[86] , B. Polichtchouk[112] , N. Poljak[129] , W. Poonsawat[114] , A. Pop[78] , S. Porteboeuf-Houssais[70] , J. Porter[74] , J. Pospisil[83] , S.K. Prasad[4] , R. Preghenella[105 ,36] , F. Prino[111] , C.A. Pruneau[135] , I. Pshenichnov[56] , M. Puccio[111] , G. Puddu[25] , P. Pujahari[135] , V. Punin[99] , J. Putschke[135] , H. Qvigstad[22] , A. Rachevski[110] , S. Raha[4] , S. Rajput[90] , J. Rak[123] , A. Rakotozafindrabe[15] , L. Ramello[32] , R. Raniwala[91] , S. Raniwala[91] , S.S. Räsänen[46] , B.T. Rascanu[53] , D. Rathee[87] , K.F. Read[125] , J.S. Real[71] , K. Redlich[77] , R.J. Reed[135] , A. Rehman[18] , P. Reichelt[53] , F. Reidt[93 ,36] , X. Ren[7] , R. Renfordt[53] , A.R. Reolon[72] , A. Reshetin[56] , F. Rettig[43] , J.-P. Revol[12] , K. Reygers[93] , V. Riabov[85] , R.A. Ricci[73] , T. Richert[34] , M. Richter[22] , P. Riedler[36] , W. Riegler[36] , F. Riggi[29] , C. Ristea[62] , A. Rivetti[111] , E. Rocco[57] , M. Rodríguez Cahuantzi[2] , A. Rodriguez Manso[81] , K. Røed[22] , E. Rogochaya[66] , D. Rohr[43] , D. Röhrich[18] , R. Romita[124] , F. Ronchetti[72] , L. Ronflette[113] , P. Rosnet[70] , A. Rossi[36 ,30] , F. Roukoutakis[88] , A. Roy[49] , C. Roy[55] , P. Roy[101] , A.J. Rubio Montero[10] , R. Rui[26] , R. Russo[27] , E. Ryabinkin[100] , Y. Ryabov[85] , A. Rybicki[117] , S. Sadovsky[112] , K. Šafařík[36] , B. Sahlmuller[53] , P. Sahoo[49] , R. Sahoo[49] , S. Sahoo[61] , P.K. Sahu[61] , J. Saini[132] , S. Sakai[72] , M.A. Saleh[135] , C.A. Salgado[17] , J. Salzwedel[20] , S. Sambyal[90] , V. Samsonov[85] , X. Sanchez Castro[55] , L. Šándor[59] , A. Sandoval[64] , M. Sano[128] , G. Santagati[29] , D. Sarkar[132] , E. Scapparone[105] , F. Scarlassara[30] , R.P. Scharenberg[95] , C. Schiaua[78] , R. Schicker[93] , C. Schmidt[97] , H.R. Schmidt[35] , S. Schuchmann[53] , J. Schukraft[36] , M. Schulc[40] , T. Schuster[137] , Y. Schutz[113 ,36] , K. Schwarz[97] , K. Schweda[97] , G. Scioli[28] , E. Scomparin[111] , R. Scott[125] , K.S. Seeder[120] , J.E. Seger[86] , Y. Sekiguchi[127] , I. Selyuzhenkov[97] , K. Senosi[65] , J. Seo[67 ,96] , E. Serradilla[10 ,64] , A. Sevcenco[62] , A. Shabanov[56] , A. Shabetai[113] , O. Shadura[3] , R. Shahoyan[36] , A. Shangaraev[112] , A. Sharma[90] , N. Sharma[61 ,125] , K. Shigaki[47] , K. Shtejer[9 ,27] , Y. Sibiriak[100] , S. Siddhanta[106] , K.M. Sielewicz[36] , T. Siemiarczuk[77] , D. Silvermyr[84 ,34] , C. Silvestre[71] , G. Simatovic[129] , G. Simonetti[36] , R. Singaraju[132] , R. Singh[79] , S. Singha[79 ,132] , V. Singhal[132] , B.C. Sinha[132] , T. Sinha[101] , B. Sitar[39] , M. Sitta[32] , T.B. Skaali[22] , M. Slupecki[123] , N. Smirnov[137] , R.J.M. Snellings[57] , T.W. Snellman[123] , C. Søgaard[34] , R. Soltz[75] , J. Song[96] , M. Song[138] , Z. Song[7] , F. Soramel[30] , S. Sorensen[125] , M. Spacek[40] , E. Spiriti[72] , I. Sputowska[117] , M. Spyropoulou-Stassinaki[88] , B.K. Srivastava[95] , J. Stachel[93] , I. Stan[62] , G. Stefanek[77] , M. Steinpreis[20] , E. Stenlund[34] , G. Steyn[65] , J.H. Stiller[93] , D. Stocco[113] , P. Strmen[39] , A.A.P. Suaide[120] , T. Sugitate[47] , C. Suire[51] , M. Suleymanov[16] , R. Sultanov[58] , M. Šumbera[83] , T.J.M. Symons[74] , A. Szabo[39] , A. Szanto de Toledo[120 ,i] , I. Szarka[39] , A. Szczepankiewicz[36] , M. Szymanski[134] , J. Takahashi[121] , N. Tanaka[128] , M.A. Tangaro[33] , J.D. Tapia Takaki[ii,51] , A. Tarantola Peloni[53] , M. Tariq[19] , M.G. Tarzila[78] , A. Tauro[36] , G. Tejeda Muñoz[2] , A. Telesca[36] , K. Terasaki[127] , C. Terrevoli[30 ,25] , B. Teyssier[130] , J. Thäder[74 ,97] , D. Thomas[118] , R. Tieulent[130] , A.R. Timmins[122] , A. Toia[53] , S. Trogolo[111] , V. Trubnikov[3] , W.H. Trzaska[123] , T. Tsuji[127] , A. Tumkin[99] , R. Turrisi[108] , T.S. Tveter[22] , K. Ullaland[18] , A. Uras[130] , G.L. Usai[25] , A. Utrobicic[129] , M. Vajzer[83] , M. Vala[59] , L. Valencia Palomo[70] , S. Vallero[27] , J. Van Der Maarel[57] , J.W. Van Hoorne[36] , M. van Leeuwen[57] , T. Vanat[83] , P. Vande Vyvre[36] , D. Varga[136] , A. Vargas[2] , M. Vargyas[123] , R. Varma[48] , M. Vasileiou[88] , A. Vasiliev[100] , A. Vauthier[71] , V. Vechernin[131] , A.M. Veen[57] , M. Veldhoen[57] , A. Velure[18] , M. Venaruzzo[73] , E. Vercellin[27] , S. Vergara Limón[2] , R. Vernet[8] , M. Verweij[135] , L. Vickovic[116] , G. Viesti[30 ,i] , J. Viinikainen[123] , Z. Vilakazi[126] , O. Villalobos Baillie[102] , A. Vinogradov[100] , L. Vinogradov[131] , Y. Vinogradov[99 ,i] , T. Virgili[31] , V. Vislavicius[34] , Y.P. Viyogi[132] , A. Vodopyanov[66] , M.A. Völkl[93] , K. Voloshin[58] , S.A. Voloshin[135] , G. Volpe[136 ,36] , B. von Haller[36] , I. Vorobyev[92 ,37] , D. Vranic[97 ,36] , J. Vrláková[41] , B. Vulpescu[70] , A. Vyushin[99] , B. Wagner[18] , J. Wagner[97] , H. Wang[57] , M. Wang[7 ,113] , Y. Wang[93] , D. Watanabe[128] , M. Weber[36] , S.G. Weber[97] , J.P. Wessels[54] , U. Westerhoff[54] , J. Wiechula[35] , J. Wikne[22] , M. Wilde[54] , G. Wilk[77] , J. Wilkinson[93] , M.C.S. Williams[105] , B. Windelband[93] , M. Winn[93] , C.G. Yaldo[135] , Y. Yamaguchi[127] , H. Yang[57] , P. Yang[7] , S. Yano[47] , Z. Yin[7] , H. Yokoyama[128] , I.-K. Yoo[96] , V. Yurchenko[3] , I. Yushmanov[100] , A. Zaborowska[134] , V. Zaccolo[80] , A. Zaman[16] , C. Zampolli[105] , H.J.C. Zanoli[120] , S. Zaporozhets[66] , N. Zardoshti[102] , A. Zarochentsev[131] , P. Závada[60] , N. Zaviyalov[99] , H. Zbroszczyk[134] , I.S. Zgura[62] , M. Zhalov[85] , H. Zhang[18 ,7] , X. Zhang[74] , Y. Zhang[7] , C. Zhao[22] , N. Zhigareva[58] , D. Zhou[7] , Y. Zhou[80 ,57] , Z. Zhou[18] , H. Zhu[18 ,7] , J. Zhu[113 ,7] , X. Zhu[7] , A. Zichichi[12 ,28] , A. Zimmermann[93] , M.B. Zimmermann[54 ,36] , G. Zinovjev[3] , M. Zyzak[43]

Affiliation notes

[i] Deceased

[ii] Also at: University of Kansas, Lawrence, Kansas, United States

Collaboration Institutes

[1] A.I. Alikhanyan National Science Laboratory (Yerevan Physics Institute) Foundation, Yerevan, Armenia
[2] Benemérita Universidad Autónoma de Puebla, Puebla, Mexico
[3] Bogolyubov Institute for Theoretical Physics, Kiev, Ukraine
[4] Bose Institute, Department of Physics and Centre for Astroparticle Physics and Space Science (CAPSS), Kolkata, India
[5] Budker Institute for Nuclear Physics, Novosibirsk, Russia
[6] California Polytechnic State University, San Luis Obispo, California, United States
[7] Central China Normal University, Wuhan, China
[8] Centre de Calcul de l'IN2P3, Villeurbanne, France
[9] Centro de Aplicaciones Tecnológicas y Desarrollo Nuclear (CEADEN), Havana, Cuba
[10] Centro de Investigaciones Energéticas Medioambientales y Tecnológicas (CIEMAT), Madrid, Spain
[11] Centro de Investigación y de Estudios Avanzados (CINVESTAV), Mexico City and Mérida, Mexico
[12] Centro Fermi - Museo Storico della Fisica e Centro Studi e Ricerche "Enrico Fermi", Rome, Italy
[13] Chicago State University, Chicago, Illinois, USA
[14] China Institute of Atomic Energy, Beijing, China
[15] Commissariat à l'Energie Atomique, IRFU, Saclay, France
[16] COMSATS Institute of Information Technology (CIIT), Islamabad, Pakistan
[17] Departamento de Física de Partículas and IGFAE, Universidad de Santiago de Compostela, Santiago de Compostela, Spain
[18] Department of Physics and Technology, University of Bergen, Bergen, Norway
[19] Department of Physics, Aligarh Muslim University, Aligarh, India
[20] Department of Physics, Ohio State University, Columbus, Ohio, United States
[21] Department of Physics, Sejong University, Seoul, South Korea
[22] Department of Physics, University of Oslo, Oslo, Norway
[23] Dipartimento di Elettrotecnica ed Elettronica del Politecnico, Bari, Italy
[24] Dipartimento di Fisica dell'Università 'La Sapienza' and Sezione INFN Rome, Italy
[25] Dipartimento di Fisica dell'Università and Sezione INFN, Cagliari, Italy
[26] Dipartimento di Fisica dell'Università and Sezione INFN, Trieste, Italy
[27] Dipartimento di Fisica dell'Università and Sezione INFN, Turin, Italy
[28] Dipartimento di Fisica e Astronomia dell'Università and Sezione INFN, Bologna, Italy
[29] Dipartimento di Fisica e Astronomia dell'Università and Sezione INFN, Catania, Italy
[30] Dipartimento di Fisica e Astronomia dell'Università and Sezione INFN, Padova, Italy
[31] Dipartimento di Fisica 'E.R. Caianiello' dell'Università and Gruppo Collegato INFN, Salerno, Italy
[32] Dipartimento di Scienze e Innovazione Tecnologica dell'Università del Piemonte Orientale and Gruppo Collegato INFN, Alessandria, Italy
[33] Dipartimento Interateneo di Fisica 'M. Merlin' and Sezione INFN, Bari, Italy
[34] Division of Experimental High Energy Physics, University of Lund, Lund, Sweden
[35] Eberhard Karls Universität Tübingen, Tübingen, Germany
[36] European Organization for Nuclear Research (CERN), Geneva, Switzerland
[37] Excellence Cluster Universe, Technische Universität München, Munich, Germany
[38] Faculty of Engineering, Bergen University College, Bergen, Norway
[39] Faculty of Mathematics, Physics and Informatics, Comenius University, Bratislava, Slovakia
[40] Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Prague, Czech Republic
[41] Faculty of Science, P.J. Šafárik University, Košice, Slovakia
[42] Faculty of Technology, Buskerud and Vestfold University College, Vestfold, Norway
[43] Frankfurt Institute for Advanced Studies, Johann Wolfgang Goethe-Universität Frankfurt, Frankfurt, Germany
[44] Gangneung-Wonju National University, Gangneung, South Korea
[45] Gauhati University, Department of Physics, Guwahati, India
[46] Helsinki Institute of Physics (HIP), Helsinki, Finland
[47] Hiroshima University, Hiroshima, Japan
[48] Indian Institute of Technology Bombay (IIT), Mumbai, India
[49] Indian Institute of Technology Indore, Indore (IITI), India
[50] Inha University, Incheon, South Korea

[51] Institut de Physique Nucléaire d'Orsay (IPNO), Université Paris-Sud, CNRS-IN2P3, Orsay, France
[52] Institut für Informatik, Johann Wolfgang Goethe-Universität Frankfurt, Frankfurt, Germany
[53] Institut für Kernphysik, Johann Wolfgang Goethe-Universität Frankfurt, Frankfurt, Germany
[54] Institut für Kernphysik, Westfälische Wilhelms-Universität Münster, Münster, Germany
[55] Institut Pluridisciplinaire Hubert Curien (IPHC), Université de Strasbourg, CNRS-IN2P3, Strasbourg, France
[56] Institute for Nuclear Research, Academy of Sciences, Moscow, Russia
[57] Institute for Subatomic Physics of Utrecht University, Utrecht, Netherlands
[58] Institute for Theoretical and Experimental Physics, Moscow, Russia
[59] Institute of Experimental Physics, Slovak Academy of Sciences, Košice, Slovakia
[60] Institute of Physics, Academy of Sciences of the Czech Republic, Prague, Czech Republic
[61] Institute of Physics, Bhubaneswar, India
[62] Institute of Space Science (ISS), Bucharest, Romania
[63] Instituto de Ciencias Nucleares, Universidad Nacional Autónoma de México, Mexico City, Mexico
[64] Instituto de Física, Universidad Nacional Autónoma de México, Mexico City, Mexico
[65] iThemba LABS, National Research Foundation, Somerset West, South Africa
[66] Joint Institute for Nuclear Research (JINR), Dubna, Russia
[67] Konkuk University, Seoul, South Korea
[68] Korea Institute of Science and Technology Information, Daejeon, South Korea
[69] KTO Karatay University, Konya, Turkey
[70] Laboratoire de Physique Corpusculaire (LPC), Clermont Université, Université Blaise Pascal, CNRS–IN2P3, Clermont-Ferrand, France
[71] Laboratoire de Physique Subatomique et de Cosmologie, Université Grenoble-Alpes, CNRS-IN2P3, Grenoble, France
[72] Laboratori Nazionali di Frascati, INFN, Frascati, Italy
[73] Laboratori Nazionali di Legnaro, INFN, Legnaro, Italy
[74] Lawrence Berkeley National Laboratory, Berkeley, California, United States
[75] Lawrence Livermore National Laboratory, Livermore, California, United States
[76] Moscow Engineering Physics Institute, Moscow, Russia
[77] National Centre for Nuclear Studies, Warsaw, Poland
[78] National Institute for Physics and Nuclear Engineering, Bucharest, Romania
[79] National Institute of Science Education and Research, Bhubaneswar, India
[80] Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark
[81] Nikhef, National Institute for Subatomic Physics, Amsterdam, Netherlands
[82] Nuclear Physics Group, STFC Daresbury Laboratory, Daresbury, United Kingdom
[83] Nuclear Physics Institute, Academy of Sciences of the Czech Republic, Řež u Prahy, Czech Republic
[84] Oak Ridge National Laboratory, Oak Ridge, Tennessee, United States
[85] Petersburg Nuclear Physics Institute, Gatchina, Russia
[86] Physics Department, Creighton University, Omaha, Nebraska, United States
[87] Physics Department, Panjab University, Chandigarh, India
[88] Physics Department, University of Athens, Athens, Greece
[89] Physics Department, University of Cape Town, Cape Town, South Africa
[90] Physics Department, University of Jammu, Jammu, India
[91] Physics Department, University of Rajasthan, Jaipur, India
[92] Physik Department, Technische Universität München, Munich, Germany
[93] Physikalisches Institut, Ruprecht-Karls-Universität Heidelberg, Heidelberg, Germany
[94] Politecnico di Torino, Turin, Italy
[95] Purdue University, West Lafayette, Indiana, United States
[96] Pusan National University, Pusan, South Korea
[97] Research Division and ExtreMe Matter Institute EMMI, GSI Helmholtzzentrum für Schwerionenforschung, Darmstadt, Germany
[98] Rudjer Bošković Institute, Zagreb, Croatia
[99] Russian Federal Nuclear Center (VNIIEF), Sarov, Russia
[100] Russian Research Centre Kurchatov Institute, Moscow, Russia
[101] Saha Institute of Nuclear Physics, Kolkata, India
[102] School of Physics and Astronomy, University of Birmingham, Birmingham, United Kingdom

[103] Sección Física, Departamento de Ciencias, Pontificia Universidad Católica del Perú, Lima, Peru
[104] Sezione INFN, Bari, Italy
[105] Sezione INFN, Bologna, Italy
[106] Sezione INFN, Cagliari, Italy
[107] Sezione INFN, Catania, Italy
[108] Sezione INFN, Padova, Italy
[109] Sezione INFN, Rome, Italy
[110] Sezione INFN, Trieste, Italy
[111] Sezione INFN, Turin, Italy
[112] SSC IHEP of NRC Kurchatov institute, Protvino, Russia
[113] SUBATECH, Ecole des Mines de Nantes, Université de Nantes, CNRS-IN2P3, Nantes, France
[114] Suranaree University of Technology, Nakhon Ratchasima, Thailand
[115] Technical University of Košice, Košice, Slovakia
[116] Technical University of Split FESB, Split, Croatia
[117] The Henryk Niewodniczanski Institute of Nuclear Physics, Polish Academy of Sciences, Cracow, Poland
[118] The University of Texas at Austin, Physics Department, Austin, Texas, USA
[119] Universidad Autónoma de Sinaloa, Culiacán, Mexico
[120] Universidade de São Paulo (USP), São Paulo, Brazil
[121] Universidade Estadual de Campinas (UNICAMP), Campinas, Brazil
[122] University of Houston, Houston, Texas, United States
[123] University of Jyväskylä, Jyväskylä, Finland
[124] University of Liverpool, Liverpool, United Kingdom
[125] University of Tennessee, Knoxville, Tennessee, United States
[126] University of the Witwatersrand, Johannesburg, South Africa
[127] University of Tokyo, Tokyo, Japan
[128] University of Tsukuba, Tsukuba, Japan
[129] University of Zagreb, Zagreb, Croatia
[130] Université de Lyon, Université Lyon 1, CNRS/IN2P3, IPN-Lyon, Villeurbanne, France
[131] V. Fock Institute for Physics, St. Petersburg State University, St. Petersburg, Russia
[132] Variable Energy Cyclotron Centre, Kolkata, India
[133] Vinča Institute of Nuclear Sciences, Belgrade, Serbia
[134] Warsaw University of Technology, Warsaw, Poland
[135] Wayne State University, Detroit, Michigan, United States
[136] Wigner Research Centre for Physics, Hungarian Academy of Sciences, Budapest, Hungary
[137] Yale University, New Haven, Connecticut, United States
[138] Yonsei University, Seoul, South Korea
[139] Zentrum für Technologietransfer und Telekommunikation (ZTT), Fachhochschule Worms, Worms, Germany

References

References for Chapter 1

[1] ALICE Collaboration. "The ALICE experiment at the CERN LHC". In: *JINST* 3.08 (2008), S08002. DOI: 10.1088/1748-0221/3/08/S08002.

[2] Lyndon Evans and Philip Bryant (editors). "LHC Machine". In: *JINST* 3.08 (2008), S08001. DOI: 10.1088/1748-0221/3/08/S08001.

[3] ALICE Collaboration. *Upgrade of the ALICE Experiment: Letter Of Intent*. Tech. rep. 2012. URL: http://cds.cern.ch/record/1475243.

[4] ALICE Collaboration. *Addendum of the Letter Of Intent for the Upgrade of the ALICE Experiment : The Muon Forward Tracker*. CERN-LHCC-2013-014 / LHCC-I-022-ADD-1. 2013. URL: http://cds.cern.ch/record/1592659.

[5] ALICE Collaboration. *Upgrade of the ALICE Inner Tracking System, Technical Design Report*. CERN-LHCC-2013-024 / ALICE-TDR-017. 2013. URL: http://cds.cern.ch/record/1625842/.

[6] ALICE Collaboration. *Upgrade of the ALICE Time Projection Chamber, Technical Design Report*. CERN-LHCC-2013-020 / ALICE-TDR-016. 2013. URL: https://cds.cern.ch/record/1622286/.

[7] ALICE Collaboration. *Technical Design Report for the The Muon Forward Tracker*. CERN-LHCC-2015-001 ; ALICE-TDR-018.

[8] ALICE Collaboration. *Upgrade of the ALICE Read-Out and Trigger system, Technical Design Report*. CERN-LHCC-2013-019 ; ALICE-TDR-015.

[9] F. Carena et al. "The ALICE data acquisition system". In: *Nucl. Instr. Meth. A* 741.0 (2014), pp. 130 –162. ISSN: 0168-9002. DOI: http://dx.doi.org/10.1016/j.nima.2013.12.015. URL: http://www.sciencedirect.com/science/article/pii/S0168900213016987.

References for Chapter 2

[1] ALICE Collaboration. *Upgrade of the ALICE Experiment: Letter Of Intent*. Tech. rep. 2012. URL: http://cds.cern.ch/record/1475243.

[2] ALICE Collaboration. *Addendum of the Letter of Intent for the Upgrade of the ALICE Experiment: The Muon Forward Tracker*. CERN-LHCC-2013-014 / LHCC-I-022-ADD-1. 2013. URL: http://cds.cern.ch/record/1592659/.

[3] ALICE Collaboration. *Conceptual Design Report for the Upgrade of the ALICE Inner Tracking System*. CERN-LHCC-2012-005 / LHCC-G-159. 2012. URL: http://cds.cern.ch/record/1431539/.

[4] ALICE Collaboration. *Upgrade of the ALICE Inner Tracking System, Technical Design Report*. CERN-LHCC-2013-024 / ALICE-TDR-017. 2013. URL: http://cds.cern.ch/record/1625842/.

References for Chapter 3

[1] ALICE Collaboration. *Upgrade of the ALICE Experiment: Letter Of Intent*. Tech. rep. 2012. URL: http://cds.cern.ch/record/1475243.

[2] ALICE Collaboration. *Addendum of the Letter of Intent for the Upgrade of the ALICE Experiment: The Muon Forward Tracker*. CERN-LHCC-2013-014 / LHCC-I-022-ADD-1. 2013. URL: http://cds.cern.ch/record/1592659/.

[3] ALICE Collaboration. *Upgrade of the ALICE Inner Tracking System, Technical Design Report*. CERN-LHCC-2013-024 / ALICE-TDR-017. 2013. URL: http://cds.cern.ch/record/1625842/.

[4] ALICE Collaboration. *Upgrade of the ALICE Time Projection Chamber, Technical Design Report*. CERN-LHCC-2013-020 / ALICE-TDR-016. 2013. URL: https://cds.cern.ch/record/1622286/.

[5] ALICE Collaboration. *Technical Design Report for the The Muon Forward Tracker*. CERN-LHCC-2015-001 ; ALICE-TDR-018.

[6] ALICE Collaboration. *Upgrade of the ALICE Read-Out and Trigger system, Technical Design Report*. CERN-LHCC-2013-019 ; ALICE-TDR-015.

[7] E. Kryshen and B. von Haller. *Detector readout time and event size*. ALICE-INT-2009-006. 2006. URL: https://edms.cern.ch/document/992666.

[8] E. Denes et al. for the ALICE collaboration. "Radiation Tolerance Qualification Tests of the Final Source Interface Unit for the ALICE Experiment". In: *Proceedings of the Topical workshop on electronics for particle physics in Valencia, Spain, 25-29 Sep 2006* 1.1-2 (2006), pp. 438–441. DOI: 10.5170/CERN-2007-001.438.

[9] F. Costa et al. for the ALICE collaboration. "DDL, the ALICE data transmission protocol and its evolution from 2 to 6 Gb/s". In: *JINST* 10 (2015), p. C04008. DOI: 10.1088/1748-0221/10/04/C04008.

[10] P. Moreira et al. "The GBT Project". In: *Proceedings of the Topical workshop on electronics for particle physics in Paris, France, September 2125* (2009), pp. 342–346. DOI: 10.5170/CERN-2009-006.

References for Chapter 6

[1] *PCIe specifications*. URL: https://www.pcisig.com/specifications/pciexpress/ (visited on 03/02/2015).

[2] *PCI-SIG - PCI Express 4.0 Frequently Asked Questions*. URL: https://www.pcisig.com/news_room/faqs/FAQ_PCI_Express_4.0 (visited on 09/12/2014).

[3] H. Engel and U. Kebschull. "High-level dataflow description of FPGA firmware components for online data preprocessing". In: *CBM Progress Report 2014* (2014).

[4] D. Eadline. *Low Cost/Power HPC*. URL: http://www.linux-mag.com/id/7799/ (visited on 03/02/2015).

[5] S. Anthony. *Intel unveils 72-core x86 Knights Landing CPU for exascale supercomputing*. URL: http://www.extremetech.com/extreme/171678-intel-unveils-72-core-x86-knights-landing-cpu-for-exascale-supercomputing (visited on 03/02/2015).

[6] T. Alt. *High-speed algorithms for event reconstruction in FPGAs*. Talk at the 17th Real Time Conference, Lisbon, 24-28 May 2010. URL: http://portal.ipfn.ist.utl.pt/rt2010/ConferencePresentations/3-Wednesday/8h30/04-RT2010.pdf (visited on 03/02/2015).

[7] *NUMA Bench — Overview*. URL: http://code.compeng.uni-frankfurt.de/projects/numabench (visited on 05/26/2014).

[8] NVIDIA Corp. *CUDA Parallel Programming*. 2013. URL: http://www.nvidia.com/object/cuda_home_new.html.

[9] Khronos Group. *OpenCL Standard for Parallel Programming*. 2013. URL: http://www.khronos.org/opencl/.

[10] OpenACC Standards Organization. *OpenACC 2.0 Standard*. 2013. URL: http://www.openacc-standard.org/node/297.

[11] OpenMP Architecture Review Board. *OpenMP API Specification for Parallel Programming*. 2013. URL: http://openmp.org/wp/openmp-specifications/.

[12] *Infiniband Trade Association Announces PF23 Results and First-Ever EDR Compliance Testing*. 2013. URL: http://www.infinibandta.org/content/pages.php?pg=press_room_item&rec_id=794.

[13] *Ethernet Alliance highlights Ethernet's continued journey at SC14*. 2014. URL: http://www.ethernetalliance.org/wp-content/uploads/2014/11/EA_SC14_FINAL_110514-2.pdf.

[14] *Intel Re-architects the Fundamental Building Block for High-Performance Computing*. 2014. URL: http://newsroom.intel.com/community/intel_newsroom/blog/2014/06/23/intel-re-architects-the-fundamental-building-block-for-high-performance-computing.

[15] D. Weller et al. "A HAMR Media Technology Roadmap to an Areal Density of 4 Tb/in2". In: *IEEE Trans. on Magnetics* 50.1 (2014).

[16] B. Marchon et al. "The Head-Disk Interface Roadmap to an Areal Density of 4 Tbit/in2". In: *Hindawi Publishing Corporation* (2013). DOI: 10.1155/2013/521086.

[17] Twitter. *Blobstore: Twitter's in-house photo storage system*. https://blog.twitter.com/2012/blobstore-twitter's-house-photo-storage-system. 2012.

[18] Subramanian Muralidhar et al. "f4: Facebook's Warm BLOB Storage System". In: *Proc. 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI'14)*. 2014.

[19] Brad Calder et al. "Windows Azure Storage: a highly available cloud storage service with strong consistency". In: *Proc. 23rd ACM Symposium on Operating Systems Principles (SOSP'11)*. 2011, pp. 143–157.

[20] Jianjun Chen et al. "Walnut: A Unified Cloud Object Store". In: *Proc. ACM conf. Management of Data (SIGMOD'12)*. 2012, pp. 743–754.

[21] *Scality RING*. URL: http://www.scality.com/ring (visited on 03/04/2015).

[22] *Caringo Swarm*. URL: http://www.caringo.com/products/swarm.html (visited on 03/04/2015).

[23] *NEC Hydrastor*. URL: http://www.necam.com/hydrastor (visited on 03/04/2015).

[24] *DDN WOS*. URL: http://www.ddn.com/products/object-storage-web-object-scaler-wos (visited on 03/04/2015).

[25] I. S. Reed and G. Solomon. "Polynomial Codes Over Certain Finite Fields". In: *Journal of the Society for Industrial and Applied Mathematics (SIAM)* 8.2 (1960), pp. 300–304.

[26] Leslie Lamport. "The part-time parliament". In: *ACM Transactions on Computer Systems* 16.2 (1998), pp. 133–169.

[27] Patrick Hunt et al. "ZooKeeper: Wait-free coordination for Internet-scale systems". In: *Proc. of the 2010 USENIX annual technical conference* (2010).

[28] Diego Ongaro and John Ousterhout. "In Search of an Understandable Consensus Algorithm". In: *Proc. of the 2014 USENIX Annual Technical Conference (USENIX ATC 14)*. 2014, pp. 305–319.

[29] *CEPH*. URL: http://ceph.com/ceph-storage/object-storage (visited on 03/04/2015).

[30] Sage A. Weil. "Ceph: reliable, scalable, and high-performance distributed storage". PhD thesis. University of California Santa Cruz, 2007.

[31] Peter Sobe and Peter Schumann. "A Performance Study of Parallel Cauchy Reed/Solomon Coding". In: *Proc. 27th int. conf. Architecture of Computing Systems (ARCS'14)*. 2014.

[32] Jianqiang Luo et al. "Efficient Encoding Schedules for XOR-Based Erasure Codes". In: *IEEE Transactions on Computers* 63.9 (2014).

References for Chapter 7

[1] *Boost C++ libraries*. URL: http://www.boost.org/.

[2] R. Brun. "ROOT An object oriented data analysis framework". In: *Nucl. Instr. Meth. A* 389.1-2 (Apr. 1997), pp. 81–86. ISSN: 01689002. DOI: 10.1016/S0168-9002(97)00048-X.

[3] *CMake Software build system*. URL: http://www.cmake.org/.

[4] *ZeroMQ - 0MQ*. URL: http://www.zeromq.org/.

[5] *Protocol Buffers: Googles Data Interchange Format. Documentation and open source release.* URL: https://github.com/google/protobuf/.

[6] F. Carena et al. "The ALICE data acquisition system". In: *Nucl. Instr. Meth. A* 741.0 (2014), pp. 130 –162. ISSN: 0168-9002. DOI: http://dx.doi.org/10.1016/j.nima.2013.12.015. URL: http://www.sciencedirect.com/science/article/pii/S0168900213016987.

[7] *Representational state transfer (REST)*. URL: http://en.wikipedia.org/wiki/Representational_state_transfer.

References for Chapter 8

[1] ALICE Collaboration. *Upgrade of the ALICE Time Projection Chamber, Technical Design Report*. CERN-LHCC-2013-020 / ALICE-TDR-016. 2013. URL: https://cds.cern.ch/record/1622286/.

[2] ALICE Collaboration. *Upgrade of the ALICE Inner Tracking System, Technical Design Report*. CERN-LHCC-2013-024 / ALICE-TDR-017. 2013. URL: http://cds.cern.ch/record/1625842/.

[3] I. Kisel. "Event reconstruction in the CBM experiment". In: *Nucl. Instr. Meth. A* 566.1 (2006), pp. 85–88. DOI: 10.1016/j.nima.2006.05.040.

[4] ALICE Collaboration. *Upgrade of the ALICE Read-Out and Trigger system, Technical Design Report*. CERN-LHCC-2013-019 ; ALICE-TDR-015.

[5] I. Hrivnacova et al. "The Virtual Monte Carlo". In: *CoRR* cs.SE/0306005 (2003). URL: http://arxiv.org/abs/cs.SE/0306005.

[6] W. Lukas. "Fast Simulation for ATLAS: Atlfast-II and ISF". In: *J. Phys. Conf. Ser.* 396 (2012), p. 022031.

References for Chapter 9

[1] ALICE Collaboration. *Upgrade of the ALICE Experiment: Letter Of Intent*. Tech. rep. 2012. URL: http://cds.cern.ch/record/1475243.

[2] ALICE Collaboration. *Upgrade of the ALICE Time Projection Chamber, Technical Design Report*. CERN-LHCC-2013-020 / ALICE-TDR-016. 2013. URL: https://cds.cern.ch/record/1622286/.

[3] "The ALICE high level trigger: The 2011 run experience". In: (2012), pp. 1–4. DOI: 10.1109/RTC.2012.6418366.

References for Chapter 10

[1] *OMNeT++ Discrete Event Simulator*. URL: http://www.omnetpp.org/.

[2] I.C. Legrand. "Multi-threaded, discrete event simulation of distributed computing systems". In: *Computer Physics Communications* (2001), pp. 274–285. DOI: 10.1016/S0010-4655(01)00281-8.

References for Chapter 11

[1] Z. Akopov et al (DPHEP study group). "Status Report of the DPHEP Study Group: Towards a Global Effort for Sustainable Data Preservation in High Energy Physics". In: *arXiv:1205.4667v1* (2012).

References for Chapter B

[1] Matteo Cacciari et al. "Theoretical predictions for charm and bottom production at the LHC". In: *JHEP* 1210 (2012), p. 137. DOI: 10.1007/JHEP10(2012)137. arXiv: 1205.6344 [hep-ph].

[2] R. Averbeck et al. "Reference Heavy Flavour Cross Sections in pp Collisions at $\sqrt{s}$ = 2.76 TeV, using a pQCD-Driven $\sqrt{s}$-Scaling of ALICE Measurements at $\sqrt{s}$ = 7 TeV". In: (2011). arXiv: 1107.3243 [hep-ph].

[3] ALICE Collaboration. *Upgrade of the ALICE Inner Tracking System, Technical Design Report*. CERN-LHCC-2013-024 / ALICE-TDR-017. 2013. URL: http://cds.cern.ch/record/1625842/.

[4] ALICE Collaboration. *Conceptual Design Report for the Upgrade of the ALICE Inner Tracking System*. CERN-LHCC-2012-005 / LHCC-G-159. 2012. URL: http://cds.cern.ch/record/1431539/.

[5] Betty Abelev et al. "Measurement of prompt J/ψ and beauty hadron production cross sections at mid-rapidity in pp collisions at $\sqrt{s} = 7$ TeV". In: *JHEP* 1211.CERN-PH-EP-2012-132 (2012), p. 065. DOI: 10.1007/JHEP11(2012)065. arXiv: 1205.5880 [hep-ex].

[6] Betty Bezverkhny Abelev et al. "Measurement of quarkonium production at forward rapidity in *pp* collisions at $\sqrt{s} = 7$ TeV". In: *Eur.Phys.J.* C74.8 (2014), p. 2974. DOI: 10.1140/epjc/s10052-014-2974-4. arXiv: 1403.3648 [nucl-ex].

[7] ALICE Collaboration. *Upgrade of the ALICE Experiment: Letter Of Intent*. Tech. rep. 2012. URL: http://cds.cern.ch/record/1475243.

[8] ALICE Collaboration. *Addendum of the Letter of Intent for the Upgrade of the ALICE Experiment: The Muon Forward Tracker*. CERN-LHCC-2013-014 / LHCC-I-022-ADD-1. 2013. URL: http://cds.cern.ch/record/1592659/.

[9] ALICE Collaboration. *Technical Design Report for the The Muon Forward Tracker*. CERN-LHCC-2015-001 ; ALICE-TDR-018.

References for Chapter C

[1] *JIRA Issue Bug/Tracking system*. URL: https://www.atlassian.com/software/jira.

[2] *Git Version control system*. URL: http://git-scm.com/.

[3] *Drupal Website creation tool*. URL: https://www.drupal.org/.

[4] *Doxygen source documentation tool*. URL: http://www.doxygen.org/.

[5] *CMake Software build system*. URL: http://www.cmake.org/.

[6] I. Legrand. " Multi-threaded, discrete event simulation of distributed computing systems". In: *Computer Physics Communications* 140.1-2 (2001), pp. 274–285. URL: http://www.sciencedirect.com/science/article/pii/S0010465501002818.

List of Figures

1.1 Functional flow of the O^2 computing system. 3

3.1 Detector read-out and interfaces of the O^2 system. 16
3.2 Block diagram of the Common Read-out Unit. 17

4.1 Job efficiency for various ALICE workflows running on the Grid during Run 1. 21
4.2 Data flow between components of the system. 22
4.3 O^2 processing flow. 25
4.4 Tier 0/Tier 1 processing flow. 26
4.5 Tier 2 simulation processing flow. 27
4.6 Relative share of various ALICE workflows running on the Grid during Run 1. 30

5.1 O^2 dataflow. 34
5.2 Detector data aggregation. 36
5.3 Hardware pipeline. 37
5.4 FLP-EPN network layout with four EPN subfarms. 39
5.5 FLP-EPN network layout with SEPNs. 40
5.6 Schematic outline of the reconstruction and calibration data flow. 41
5.7 Quality control and assessment general workflow. 43
5.8 CCM systems overview. 44
5.9 CCM interfaces with external systems. 45
5.10 DCS context. 46

6.1 DMA to host performance. 50
6.2 DMA throughput measurements. 51
6.3 Evolution of maximum throughput to DDRx memory over the last years. 52
6.4 Performance of the FPGA-based FastClusterFinder algorithm. 55
6.5 HLT TPC Tracker Performance. 56

6.6 Cluster finding performance on x86. 57
6.7 Throughput of the ”Add One” test of NUMAbench. 59
6.8 Code size as opposed to computational effort. 61
6.9 Conventional kernel code. 61
6.10 Performance using different number of disks in RAID set. 63
6.11 Storage performance using different block sizes. 64
6.12 Prediction of hard disks storage density. 65
6.13 Building blocks of a Lustre system. 66
6.14 Lustre performance for one client using one stream with different block sizes. 67
6.15 Lustre performance for multiple clients using 1 or 2 streams. 67
6.16 Lustre performance for multiple clients using 1 or 2 streams. 68
6.17 Lustre performance for 1 client using multiple streams. 68
6.18 SMI performance plot. 73
6.19 ZeroMQ and Boost Meta State Machine performance plot. 73
6.20 MonALISA performance plot. 74
6.21 Zabbix performance plot. 74

7.1 O^2 software ecosystem. 75
7.2 Modules of the ALFA framework and the libraries and tools used. 76
7.3 Definition of an abstract task. 78
7.4 Process base state machine. 79
7.5 Examples of parallel Activities and their associated Partitions. 80
7.6 Interaction between CCM Agents and Processes. 82
7.7 Main phases executed by read-out. 83
7.8 Data format at the level of the FLPs. 85
7.9 The Time Frame descriptor assembled on FLPs. 87
7.10 Format of the Time Frames between FLPs and EPNs. 88
7.11 Quality control and assessment general design. 90
7.12 Event Display design. 92
7.13 The DCS Data Collector and DCS Access Points. 94

8.1 Schematic outline of the reconstruction and calibration data flow. 96
8.2 Schematic outline of the reconstruction and calibration data flow. 100
8.3 Schematic diagram of the seeding procedure and t_0 estimation. 104

8.4 Probability of at least one pile-up collision. 107

9.1 Data compression factor versus raw data size. 112
9.2 Projection of a pp event on the transverse plane of the TPC. 113

10.1 Throughput between two machines connected with 40 Gb/s Ethernet. 120
10.2 Link speed - 4 EPN farms. 122
10.3 Link speed - Super-EPNs. 122
10.4 Bisection traffic - 4 EPN farms. 123
10.5 Bisection traffic - Super-EPNs. 123
10.6 Buffers size - 4 EPN farms. 123
10.7 Buffers size - Super-EPNs. 124
10.8 Total data stored for a month of data taking similar to Run 1. 124
10.9 System scalability of the network layout 2 at up to 80 kHz. 125
10.10 System scalability of the network layout 3 at up to 140 kHz. 125

11.1 O^2 Project schedule. 130

B.1 Relative uncertainty for the scaling factor of the cross section of D and B mesons. 143

C.1 The evaluation procedure 146

List of Tables

2.1 Estimated S/ev, S/B and B'/ev. 10

2.2 Summary of integrated luminosity requirements. 10

2.3 ALICE running scenario. 11

3.1 Data rates per detector. 15

3.2 Detector links and read-out boards used to transfer the data from the detectors to the O^2 system [6]. 17

3.3 Number of simulated events for different systems. 18

3.4 Grid resources evolution. 19

4.1 Components. 22

4.2 Data Types. 23

4.3 Sizes. 23

4.4 Number of reconstructed collisions and storage requirements. 24

4.5 Number of simulated events and storage requirements for different systems. 24

5.1 FLP and EPN characteristics. 38

6.1 Evolution of PCI Express. 50

6.2 Result of the PCI Express Benchmark. 58

6.3 Results of the TPC Track Finder benchmark. 60

6.4 Results of the TPC Track Fit Benchmark. 60

6.5 Result of the Matrix Multiplication Benchmark. 60

6.6 Result of the Infiniband and Ethernet network benchmarks. 62

6.7 Technology future bets. 65

6.8 Tools to implement CCM functions. 72

8.1 Summary of the different detector calibrations for TPC and ITS. 98

8.2 Summary of the different detector calibrations for TRD. 99

8.3 Summary of the different detector calibrations for TOF. 100
8.4 Summary of the different detector calibrations for MCH, MID, and MFT. 101
8.5 Summary of the different detector calibrations for FIT. 101
8.6 Summary of the different detector calibrations for HMP and ZDC. 102
8.7 Summary of the different detector calibrations for Calorimeters (EMC, PHS) and CPV. . 102
8.8 CPU requirements for processing the data. 108

9.1 Data reduction steps for the TPC. 111
9.2 Data rates for input and output of the O^2 system. 114

10.1 Number of read-out boards and FLPs per detector to O^2 system. 116
10.2 Nodes of the O^2 facility with their minimal characteristics. 117
10.3 Network ports, throughput and bandwidth. 117
10.4 Storage needs. 117
10.5 Power and cooling needs. 119

11.1 O^2 Project Computing Working Groups and their topics. 127
11.2 Institutes participating in the O^2 Project. 128
11.3 Sharing of responsibilities and human resources needed. 129
11.4 Cost estimates and spending profile. 131

B.1 Statistical uncertainties for Λ_c production measurement. 143
B.2 Statistical uncertainties for charmonium. 144