

Performance tests of *InfluxDB* database

Adam Wegrzynek

30.01.2017

The performance of *InfluxDB* database was evaluated as it is a major candidate for metrics storage in O² Monitoring sub-system.

Figure 1 presents current state of the Monitoring sub-system. Components marked with red "X" are still in evaluation phase and no preliminary solution has been proposed.

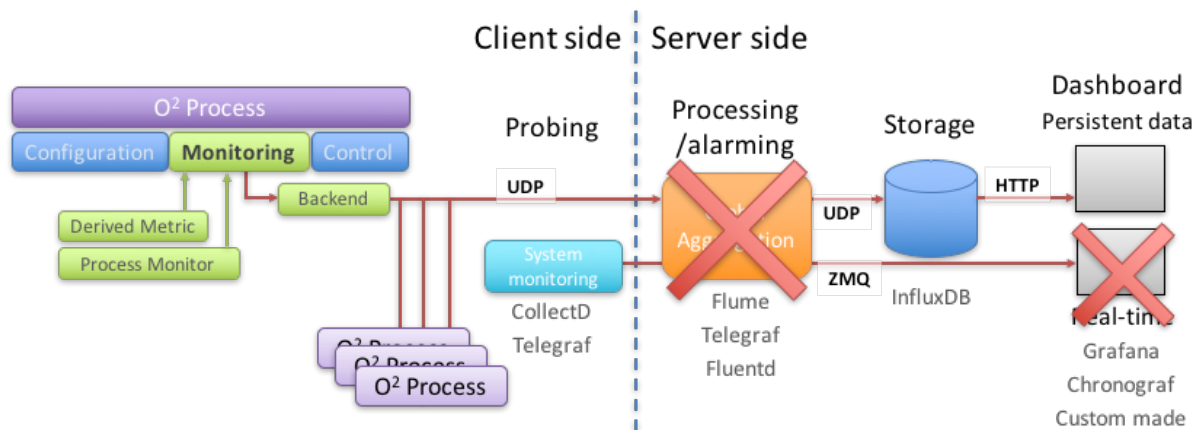


Figure 1. Current state of Monitoring sub-system

1. Requirements

It is expected that Monitoring sub-system will cope with approximately 600 000 raw metrics per second. Between data collectors and database, processing layer is foreseen that will discard and aggregate significant number of metrics. Therefore, exact metric rate that will reach the database is still not estimated.

2. Benchmark

The client side benchmark used in the test is based on Monitoring library (which currently is part of FLP Prototype). The benchmark sends the same metric over and over again with a given rate. The exact source code and configuration file can be found on GitLab (<https://gitlab.cern.ch/alice-o2-flp-prototype/hackathon/tree/master/Monitoring/examples>): *6-Benchmark.cxx* and *SampleConfig.ini*.

3. Environment

InfluxDB instances were installed on two nodes with following software and hardware configurations – see Table 1.

aidrefsrv12 (HDD)	aido2mon (SSD)
E5-2690	E5-2670 v3
64 GB RAM	64 GB RAM
40 GbE Chelsio T580	10 GbE Intel X520
HDD storage	SSD storage
InfluxDB v1.1	
CentOS 7 (3.10.0-514.6.1.el7.x86_64)	

Table 1. Software and hardware configurations

The client side benchmark was launched from *aidrefsrv10* node which hardware configuration is the same as *aidrefsrv12*.

To increase the performance each InfluxDB instance had 3 UDP endpoints opened. In addition, UDP receive buffer size was increased to 25 MB.

4. HTTP vs UDP

InfluxDB may receive metrics at either HTTP or UDP endpoint. Figure 2 shows how many metrics were stored in the database during 120 seconds run with one of given rates using HTTP and UDP. On the left side of the figure there's ideal, expected numbers of values (120s x rate). It can be clearly observed that HTTP is limited by 5 kHz rate. UDP can get as far as ~100 kHz with only few (<1%) packet drops. More detailed measurements are presented in following section.

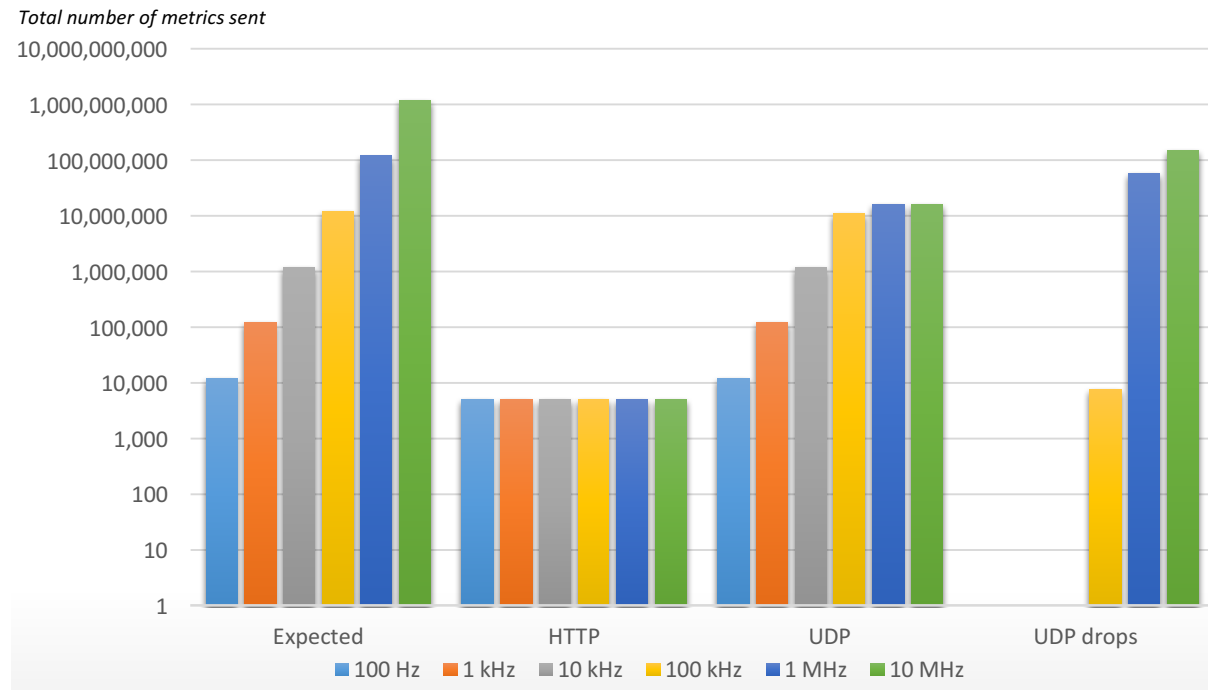


Figure 2. InfluxDB – HTTP vs UDP

5. UDP Performance

As performance of HTTP endpoints is not satisfactory more precise tests were performed only with UDP.

Before each test all metrics in database are erased. Then, the client side benchmark is executed. The script launches multiple instances of the benchmark. Each instance sends metrics to one of 3 UDP endpoints. The script counts number of UDP packets sent during the test, which is almost equal to the number of metrics sent by the benchmark (number of UDP packets sent by operating system during the test is negligible). This value is compared to the total number of metrics present in the database.

Above procedure allows to calculate how many metrics sent by the benchmark were actually stored in the database – see Figure 3.

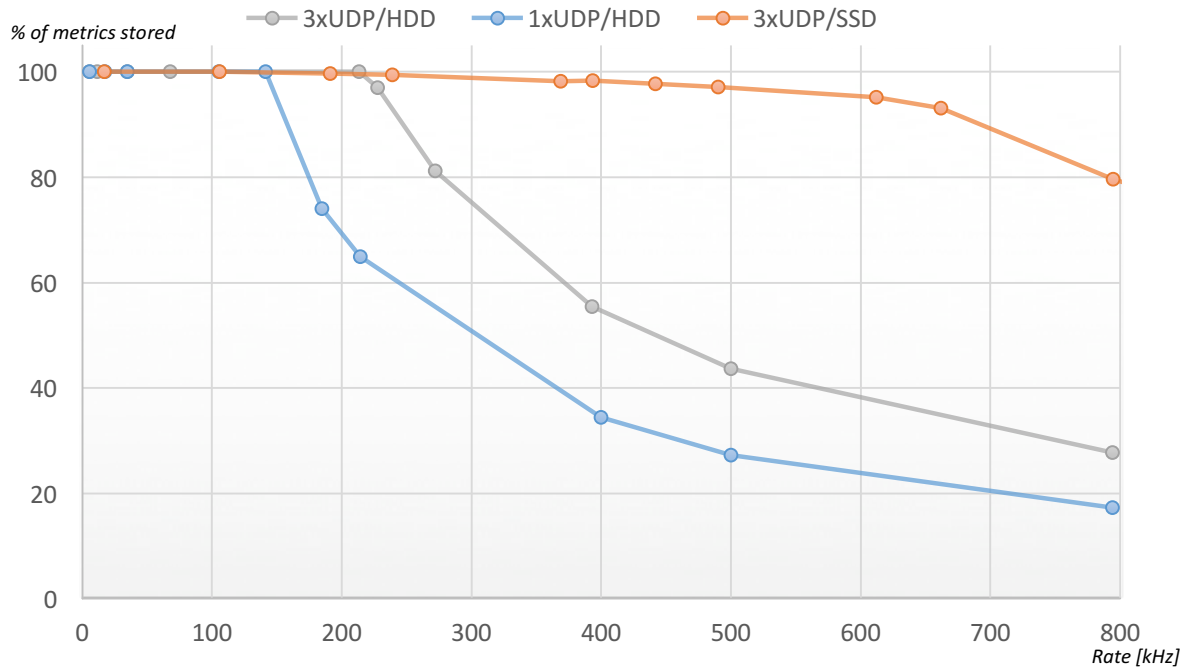


Figure 3. UDP performance - % of metrics stored in function of metric rate

Setup with magnetic disk (HDD) reached 142 kHz with single UDP endpoint and 216 kHz with 3 UDP endpoints. Further endpoint increase (over 3) does not have any impact on results. As expected, at some point for higher rates sharp drop is observed.

Setup with flash disk (SSD) behaves very different. The percentage of stored metrics decreases very slowly, what has not been fully understood. With 300kHz rate 1% of metrics are lost, with 480 kHz 3%.

6. Conclusions

InfluxDB meets initial requirements of Monitoring sub-system. Either 1 or 2 InfluxDB instances (depending on final metric rate) will be able to cope with the traffic.

7. Future work

New versions of InfluxDB are released almost each month, therefore they should be tested in the same setup to track performance trend.

In addition, in further stage it would be beneficial to understand which components limit the overall performance (kernel and hard drive are the main suspects).