Monitoring of the new ALICE Online-Offline computing system

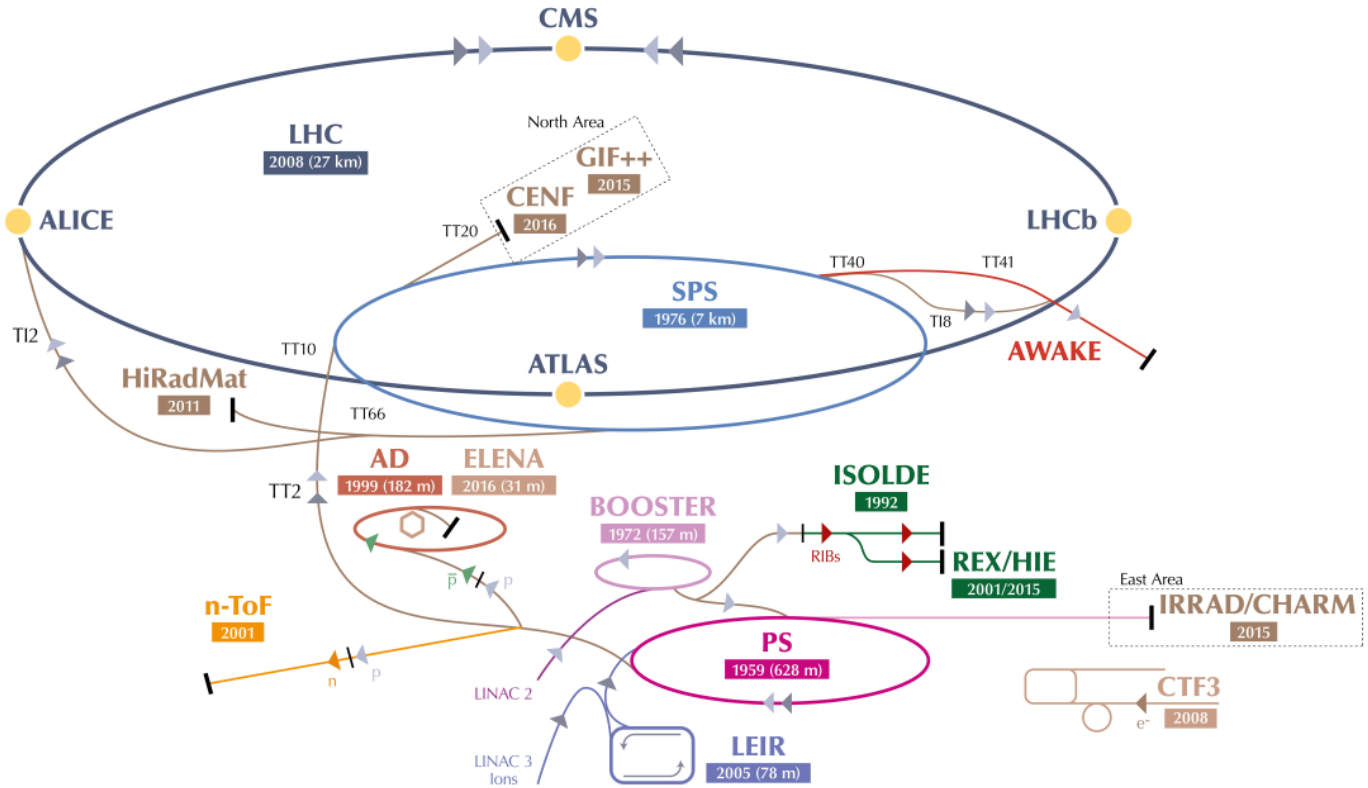
Adam Wegrzynek • CERN

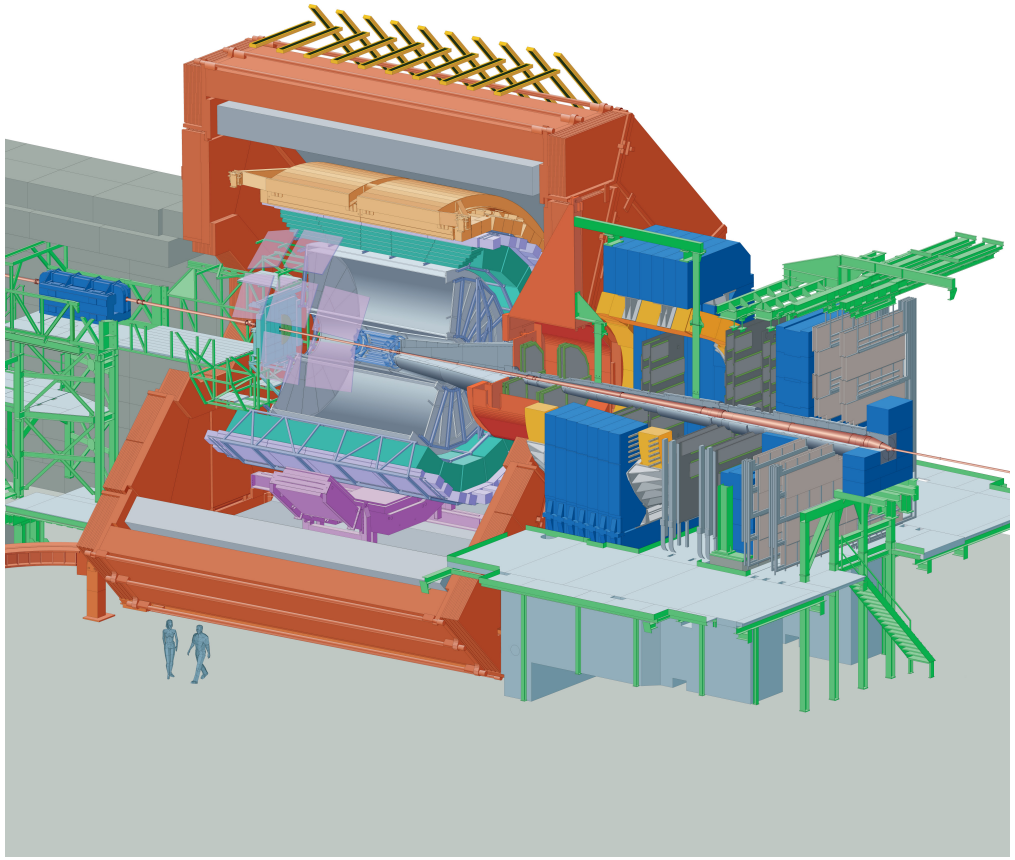
for the ALICE Collaboration

Overview

- Introduction
- Functionality
  - Requirements
  - Modular Stack solution
- Performance
  - Requirements and test procedure
  - Metric rate and latency measurements
- Conclusion

Introduction





Introduction: O^2 System

Continuous readout support

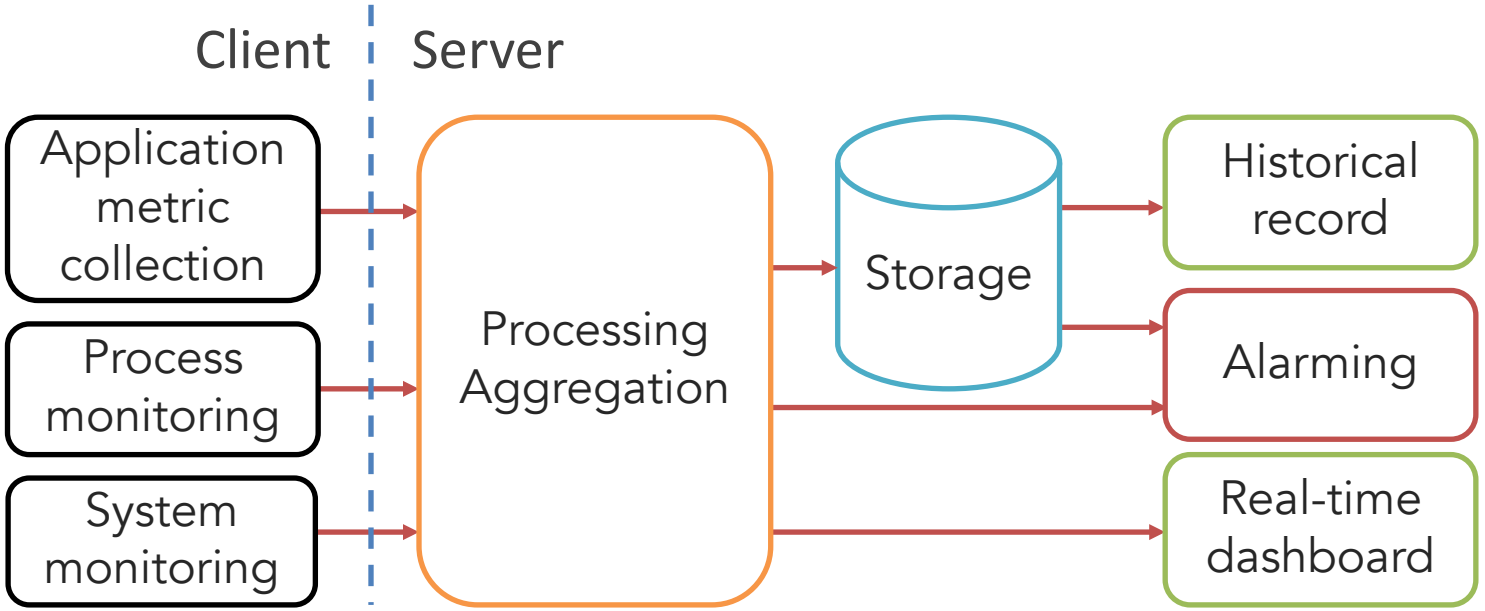
Replacement of the current Online and Offline computing systems

2 000 machines,
100 000 processes

27 Tb/s raw data
reduced to 720 Gb/s

100x larger data volume

Requirements: Functional (1)



- Application metric collection
  - Collects user defined metrics
  - Initial aggregation
- Process monitoring
  - Collect process performance metrics: CPU and memory usage, bytes received and sent, etc.

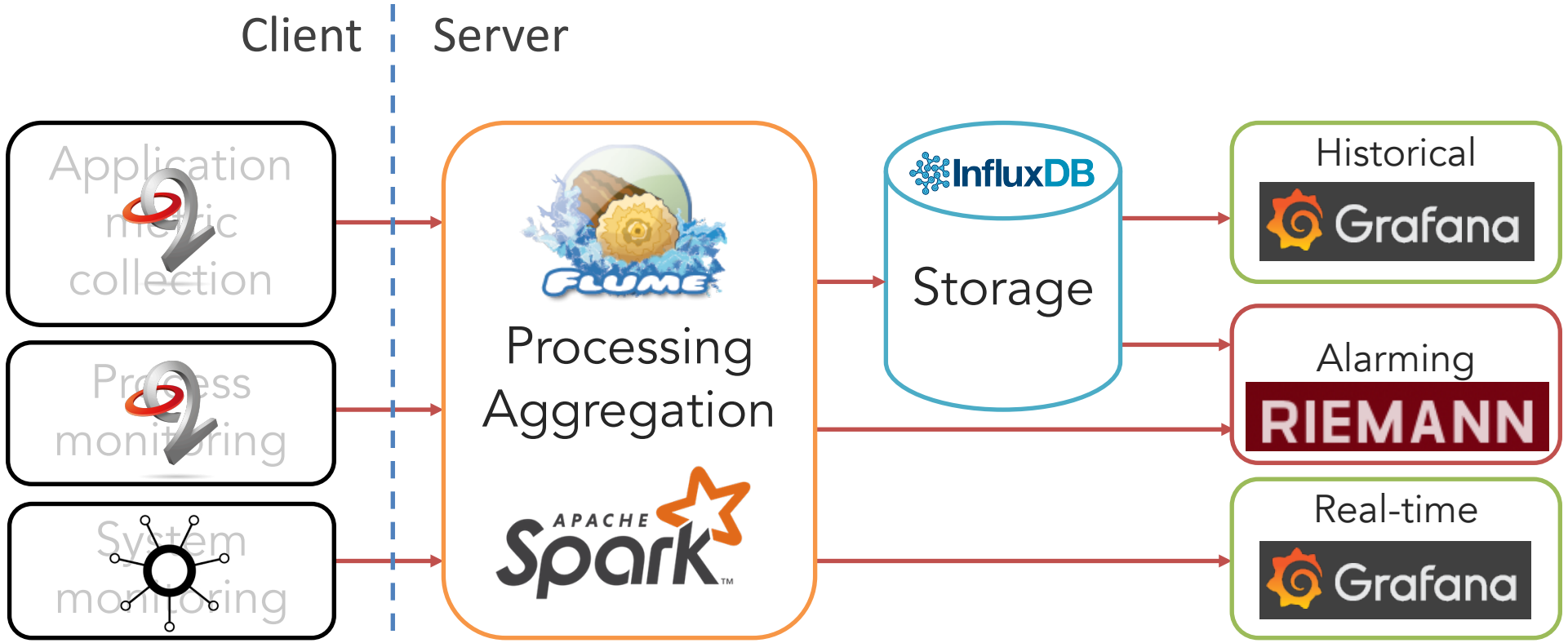
Requirements: Functional (2)

- System monitoring
  - Operating system probes to CPU, memory, network, storage…
- Metric aggregation and processing
  - Suppression
  - Enrichment
  - Aggregation
  - Correlation
- Storage
  - Archiving
  - Downsampling
- Visualization
  - Real-time
  - Historical record
- Alarming
  - Thresholds
  - Missing value
  - Custom logic

Modular stack

1. Monitoring library
   - Application metric collection
   - Process monitoring
   - Includes benchmark
   - GitHub page
2. collectd
   - Collects system performance metrics
3. Flume
   - Moves metrics from multiple sources into centralized store
   - Basic processing
4. Apache Spark
   - In memory data processing
5. InfluxDB
   - Time series database
   - Continuous queries
   - Retention policy
6. Grafana
   - Time series visualization tool
7. RIEMANN
   - Alarming

Modular stack mapped on functional architecture



Requirements: Performance

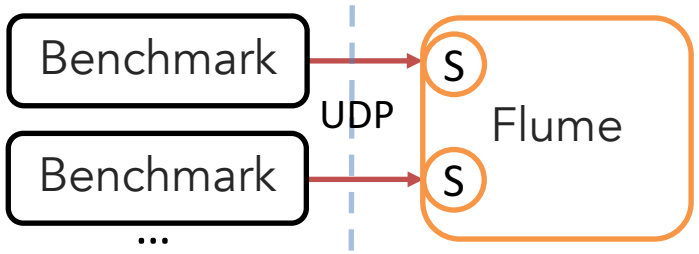
- Capable of handling O^2 monitoring traffic at 600 kHz
- Scalable to >> 600 kHz
- Handle at least 100 000 sources
- Introduce latency no higher than
  - 500 ms to processing layer
  - 1000 ms to presentation layer
- Impose low storage size per measurement

Performance measurements

- Reference setup
  - 3 backend machines
    - E5-2640 v3, 40 GbE, HDD/SDD
  - 40 client machines
  - Benchmark based on Monitoring library
  - Ansible recipes to run tests semi-automatically
- Processing scenarios
  - Passthrough – forwards metric to the storage
  - Edit a metric – modify one of the metric's fields
  - Aggregation – aggregates metrics of the same origin

Metric rate: Benchmark-Flume

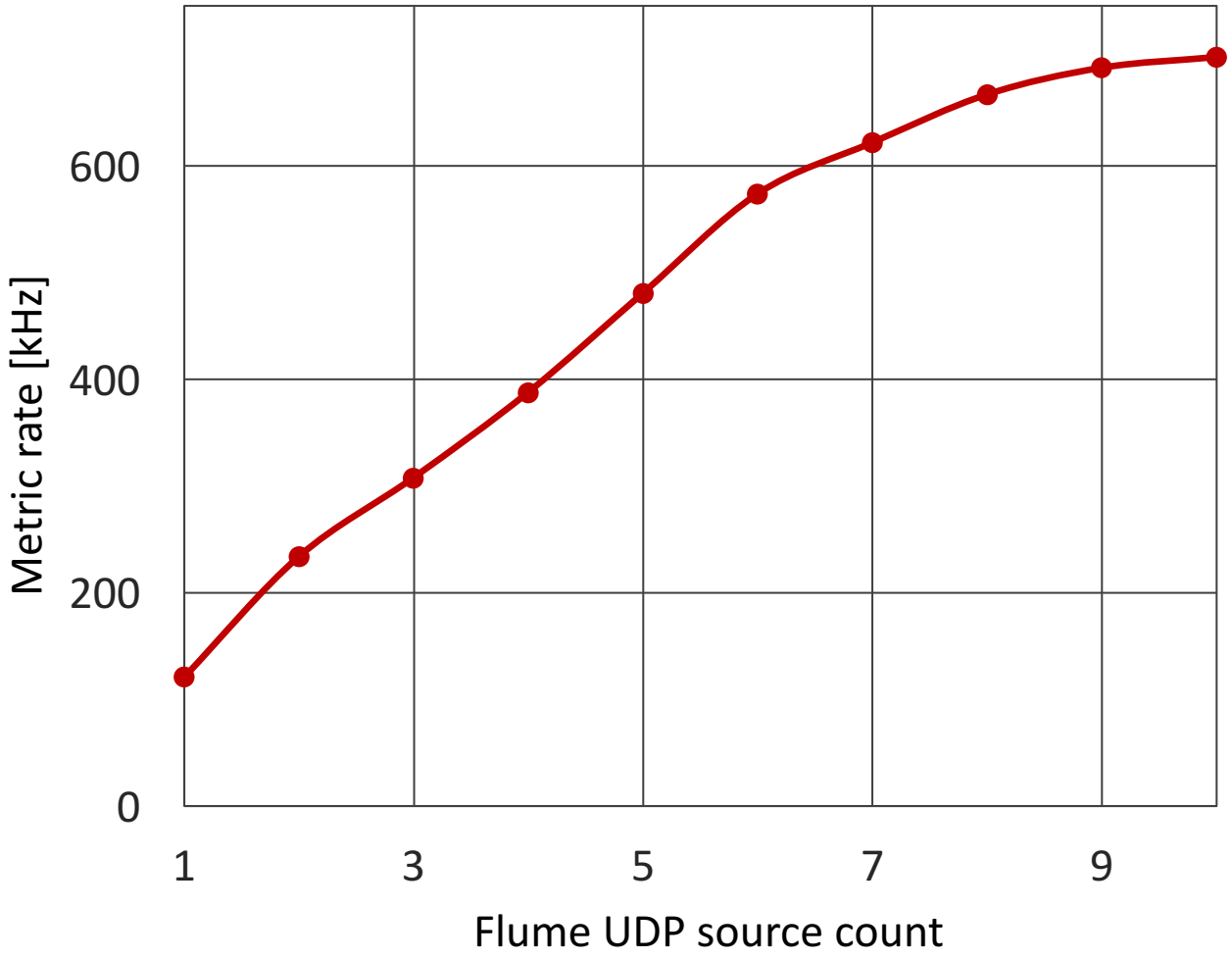
CONFIGURATION



COMMENTS

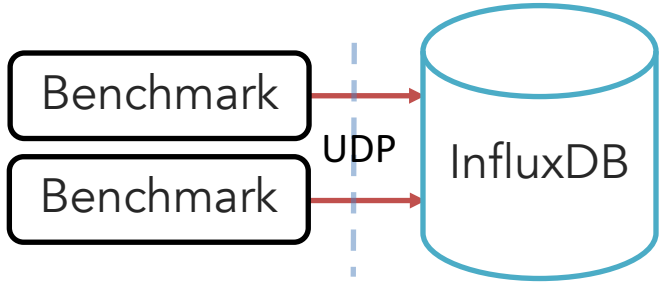
- Initially linear increase
- Saturation at 630 kHz
- Multiple values per UDP packet possible

MEASUREMENT



Metric rate: Benchmark-InfluxDB

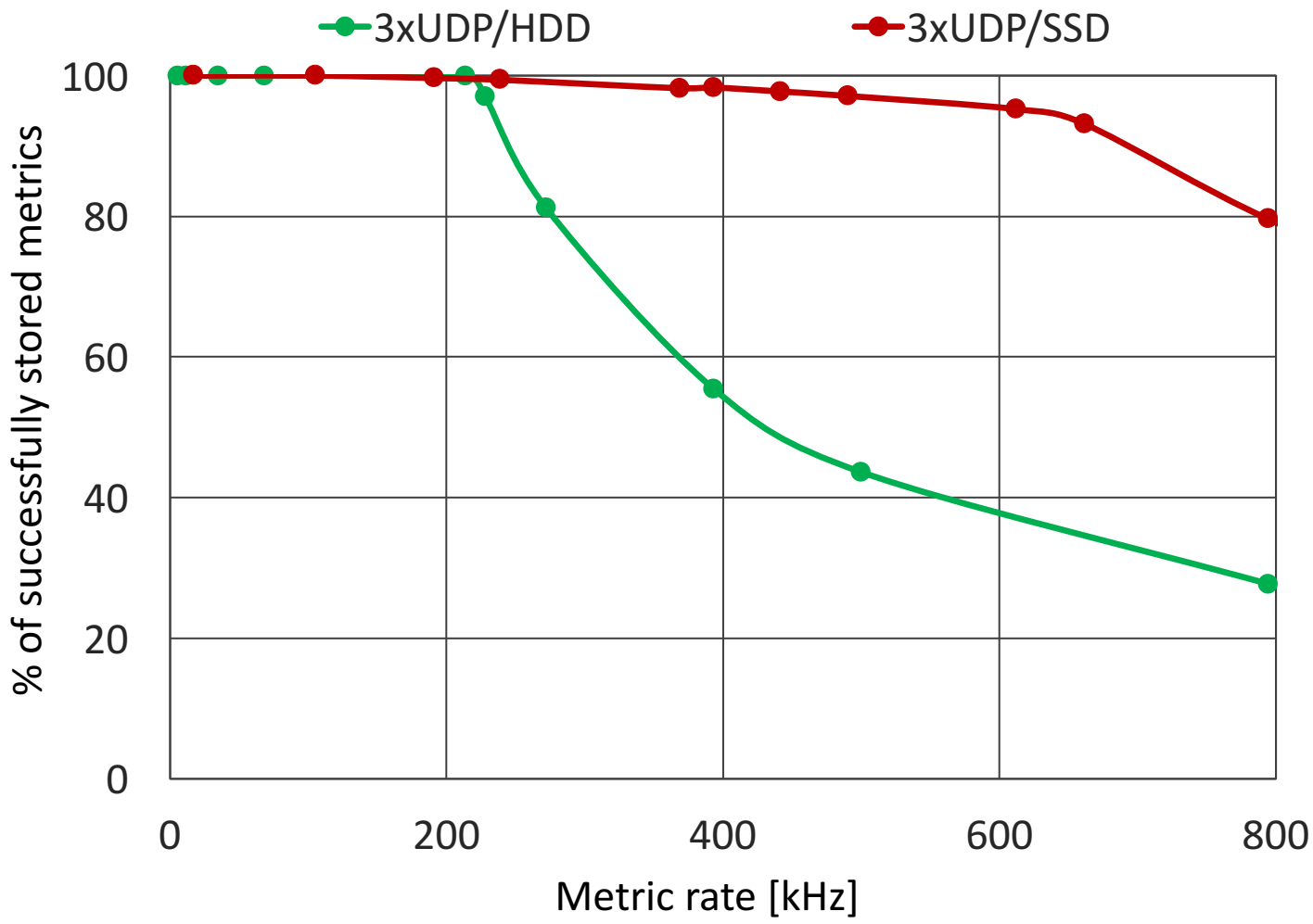
CONFIGURATION



COMMENTS

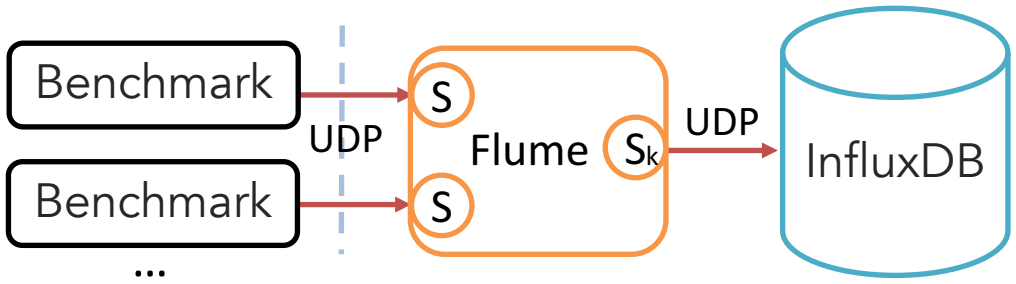
- HDD
  - Drop at 216 kHz with 3 UDP listeners
- SSD
  - Gradual decrease (not fully understood behavior)

MEASUREMENT



Metric rate: Benchmark-Flume-InfluxDB

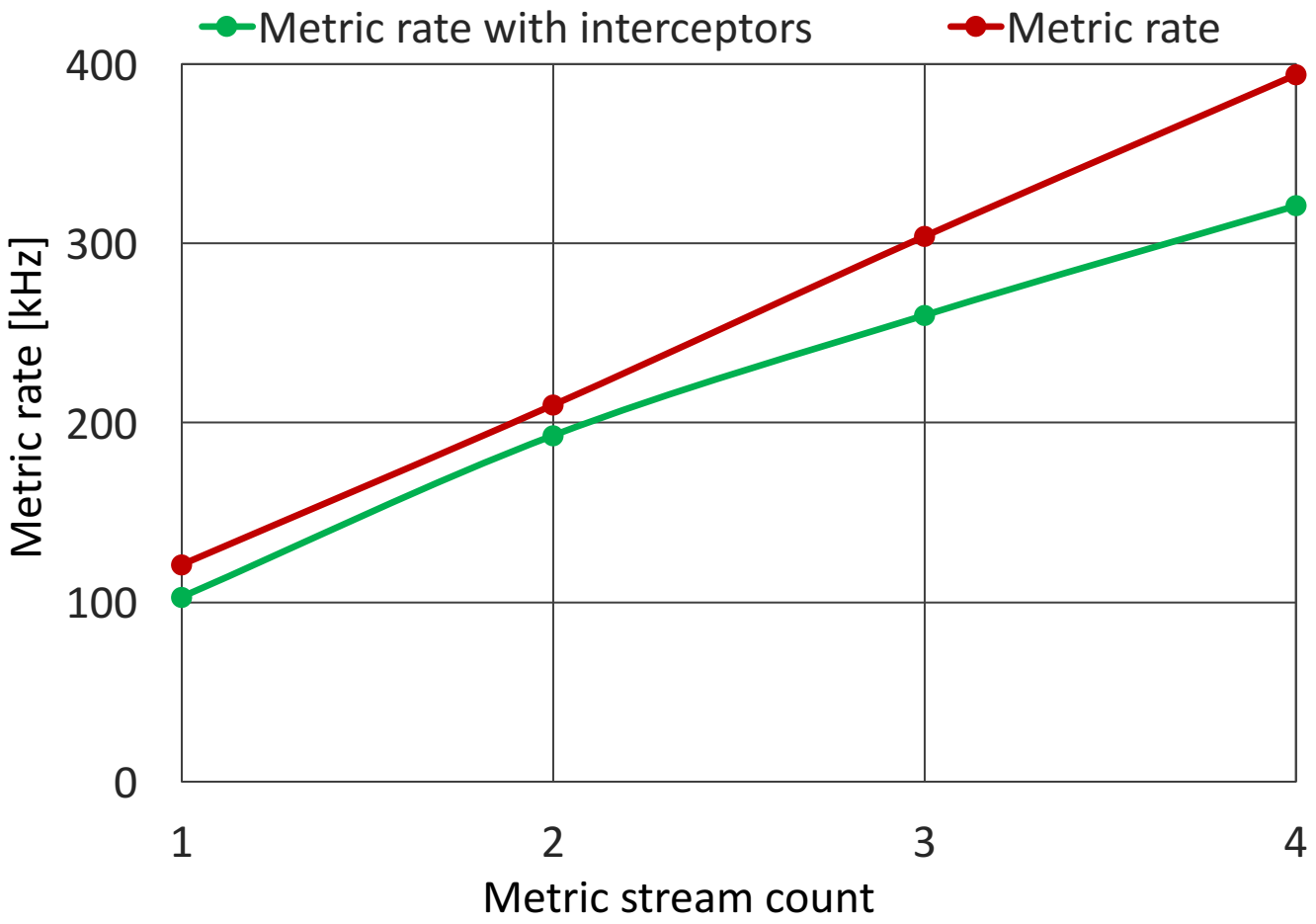
CONFIGURATION



COMMENTS

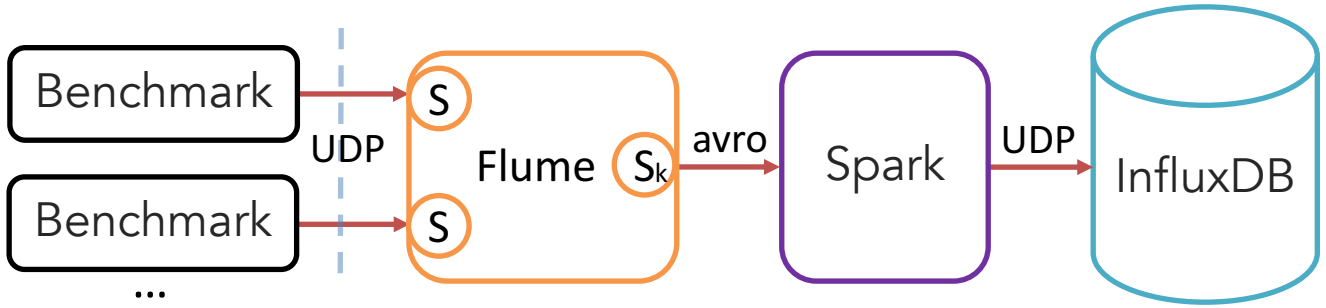
- Metric stream consists of dedicated
  - Benchmark instance
  - Flume source and sink
  - InfluxDB listener
- 4 streams – 400 kHz

MEASUREMENT



Metric rate: Benchmark-Flume-Spark-InfluxDB

CONFIGURATION



COMMENTS

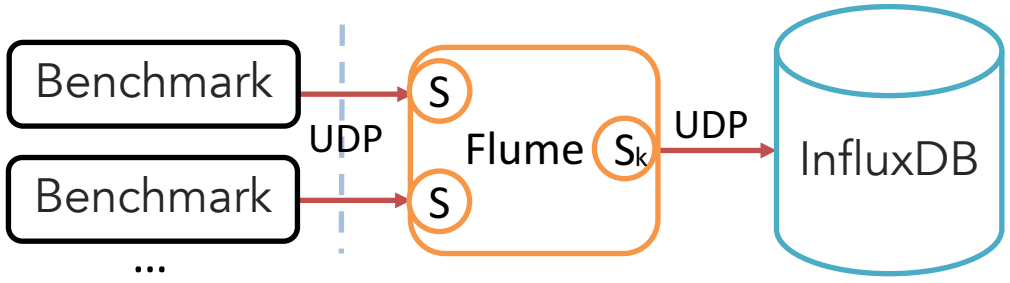
- Multiple instances of Spark required

MEASUREMENT

- Pass through
  - 207 kHz
- Batch processing – average value algorithm over 1000 ms time period
  - 180 kHz

Latency: Benchmark-Flume-InfluxDB

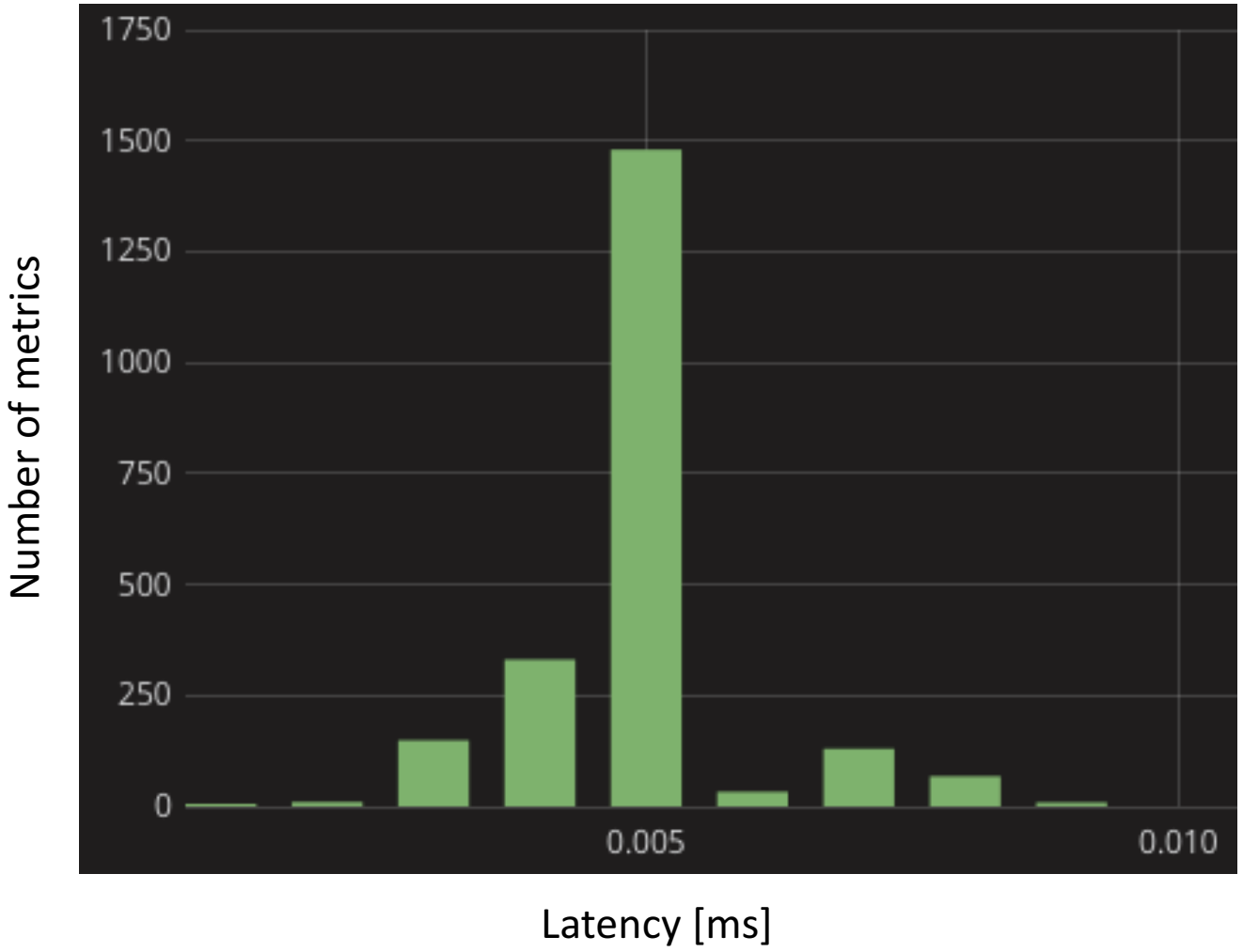
CONFIGURATION



COMMENTS

- The visualization is not part of the measurement as real time updates has not yet been implemented
- Most of metrics transmitted within 5 μs

MEASUREMENT



Issue with NUMA and interrupts

- Problem definition
  - Periodical performance drops when running tests
- Study
  - NIC interrupts decrease performance of an application running at the same core because of large number of context switches
  - In addition, the Linux scheduler (3.10.0-514.26.2.el7.x86_64) does not take into account influence of network interrupts when moving processes between CPU cores
- Solution
  - Bind application to dedicated CPU core of the same NUMA node as interrupts

Conclusions

- Modular Stack is capable of monitoring the future O^2 system
  - Meets functional requirements
  - Reaches 600 kHz metric rate
  - Provides low latency

Future work

- Include visualization layer in the latency measurement
- Implement real processing scenarios in Spark
- Create feedback from alarming to control module