Efficient Time Frame Building for Online Data Reconstruction in ALICE Experiment

Alexey Rybalchenko
(Darmstadt University of Applied Sciences, GSI Darmstadt)
On behalf of the ALICE Collaboration

21st International Conference on Computing in High Energy and Nuclear Physics
April 14, 2015, Okinawa, Japan

h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Challenges for the ALICE Online-Offline Computing System in Run 3 (2020)

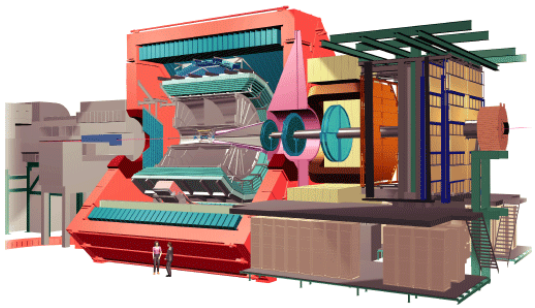
- Physics objectives require high statistics due to low cross sections. Very small signal-to-background ratio makes classical trigger system impossible.

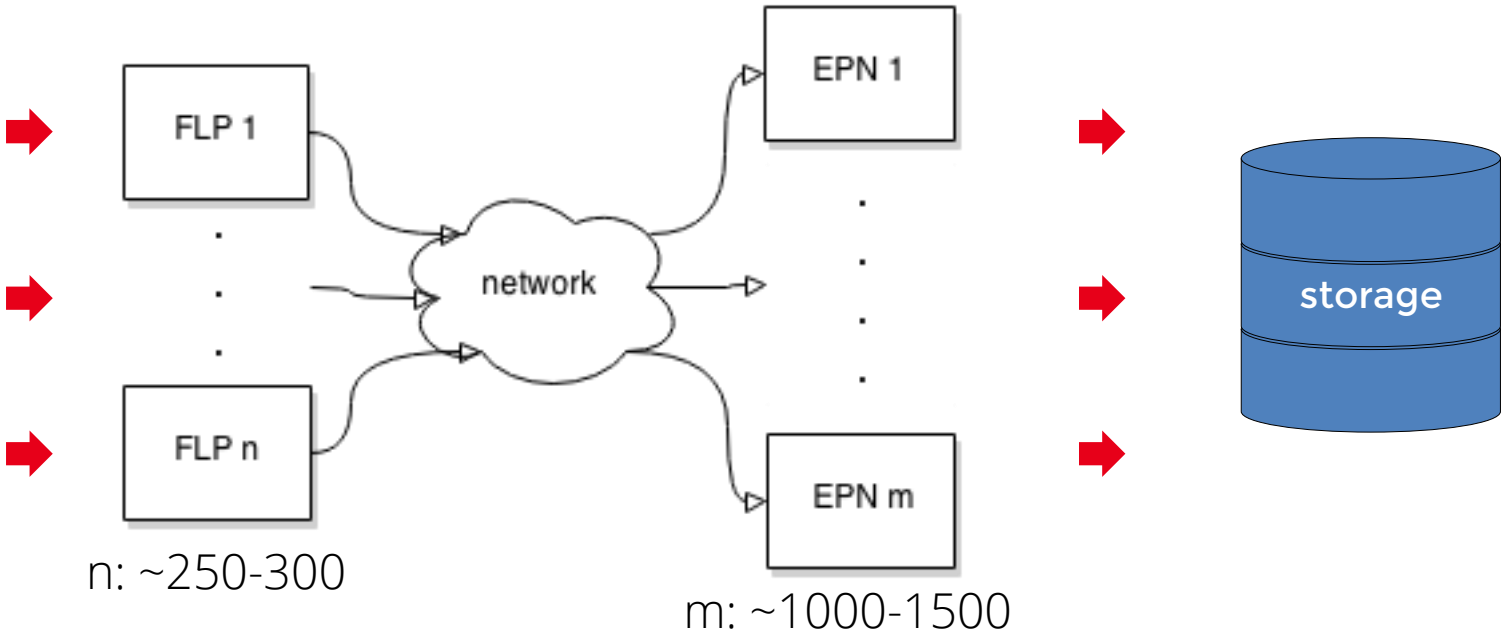
- Higher interaction rate and upgraded detector systems. Main detectors (ITS, TPC) will operate in continuous mode.

➡ Very large detector output data rate of **~1.2 TByte/s**.

➡ **Partial/full reconstruction** for data reduction necessary to meet storage bandwidth requirements.

Outline of the data flow for Online-Offline Computing System in Run 3

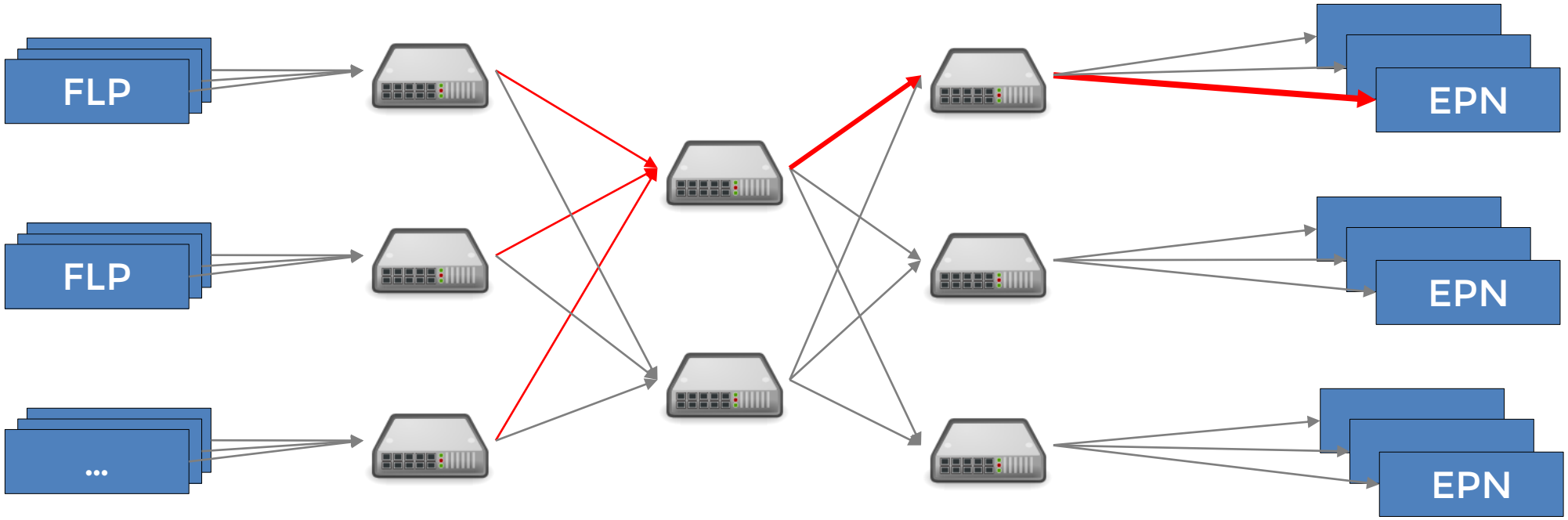




- **FLPs (First Level Processor)** receive data from the detector readout, preprocess it, chop it into manageable pieces (time frames) and send it out to EPNs.
- **EPNs (Event Processing Node)** collect sub-time frames from **all** FLPs to build a full time frame for reconstruction (on the EPN nodes).

The network design should take emerging technologies into account and thus be finalized at a later stage.

Challenge for the underlying network between FLPs and EPNs



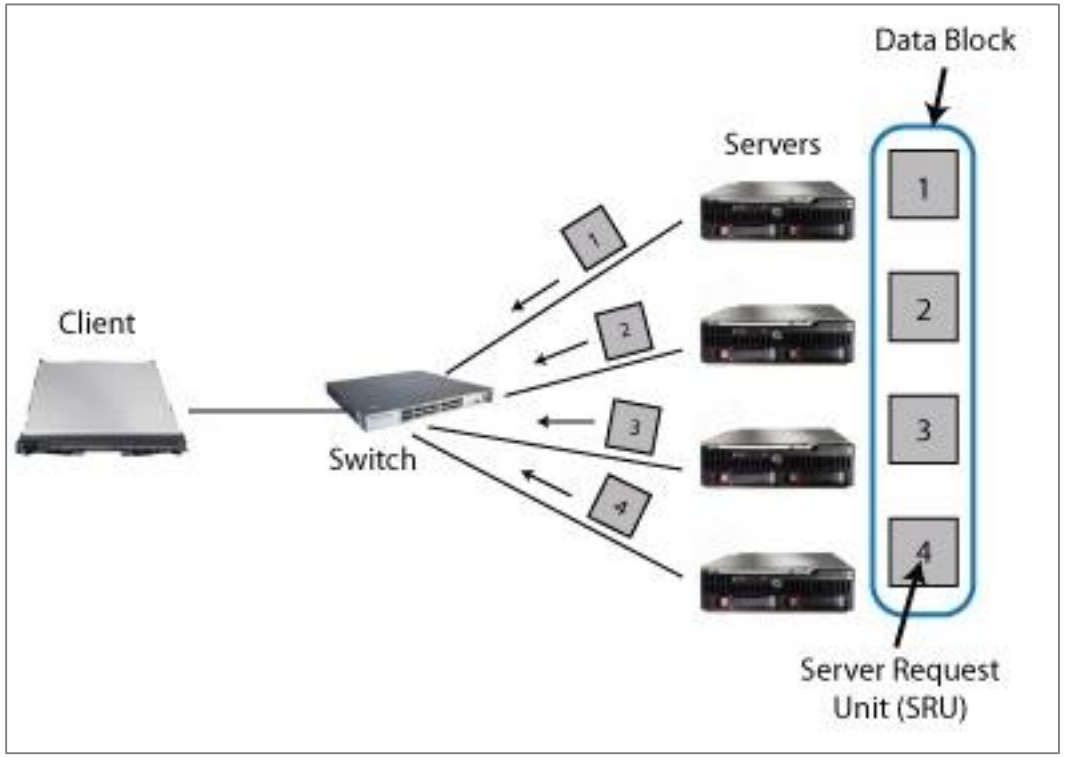
- One EPN requires sub-time frames from **all** FLPs for reconstruction.
- The size of one time frame is **11 GB**.
- FLPs attempt to transfer to the same EPN **at the same time**.
- This requires each link in the network to handle data from **all** senders simultaneously.
- Related problem: when using TCP on Ethernet, this pattern can lead to **throughput collapse** (TCP Incast).
- FLPs should operate at maximum performance - **avoid/minimize synchronization**.

TCP Incast in Big Data Applications

When many packet losses occur, TCP issues a retransmission timeout (RTO), lasting at least 200ms.

Many solutions proposed to solve the TCP Incast problem:

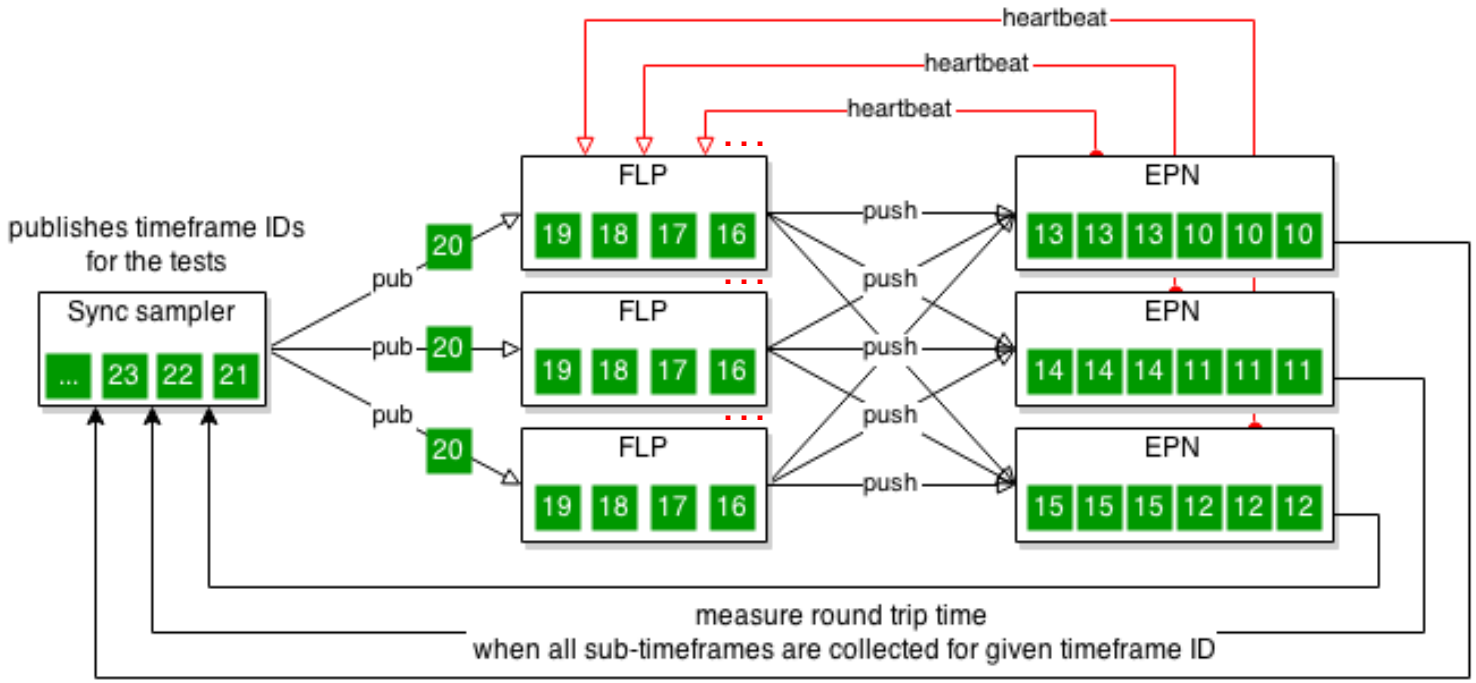
- **Network-based**: Tune the network protocol/settings. Reduce RTO value; explicit feedback mechanism; custom congestion control.
- **Application-based**: Global scheduling of the data transfers; Staggering of the data transfers.
- **$-based**: Increase buffer/capacity on the switches.



Source: http://www.pdl.cmu.edu/Incast/

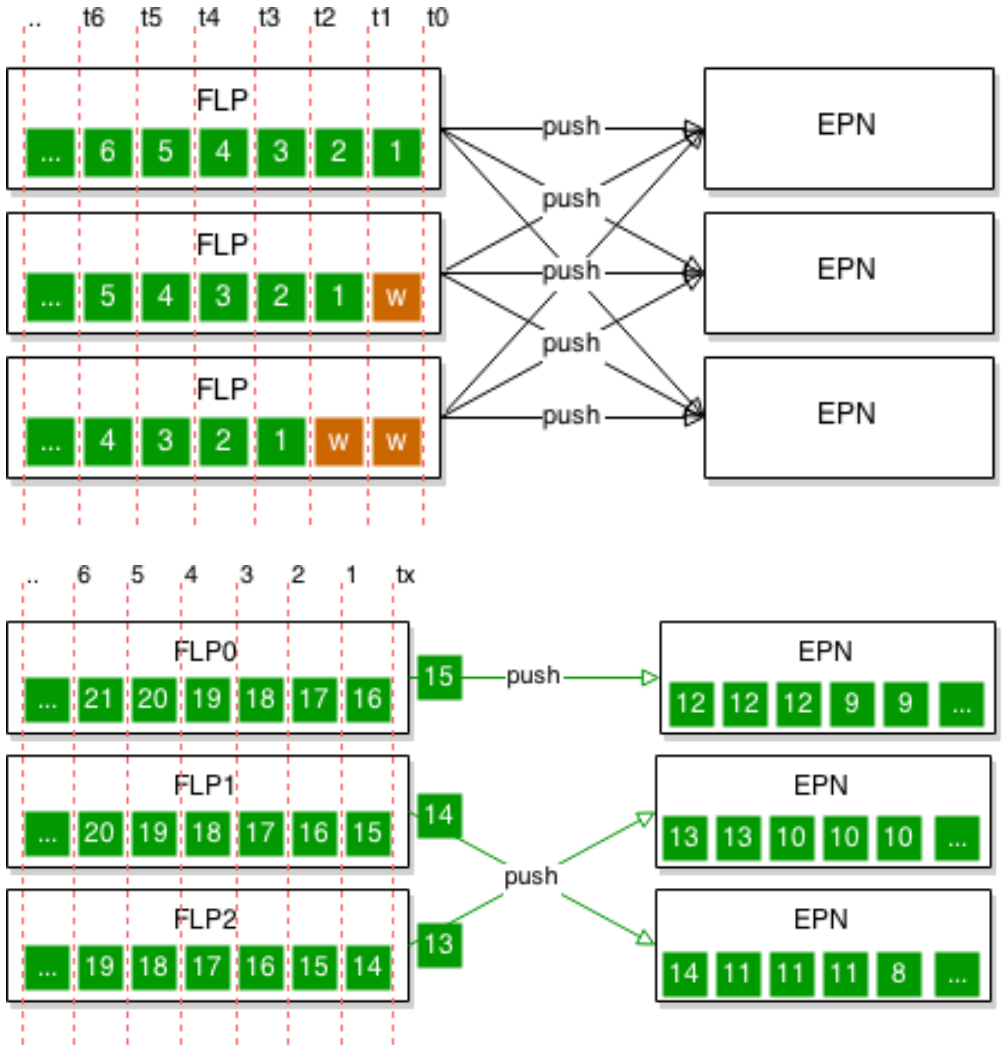
FLP2EPN Test Setup

- Sync Sampler publishes time frame IDs at configurable rate.
- FLPs generate dummy data of configurable size and distribute it to EPNs.
- FLP decides where to send the payload from the time frame ID: **TimeFrameID % numEPNs**
- EPN availability is tracked with the heartbeat received within timeout window on each FLP.
- Upon collecting sub-time frames from all FLPs, EPN sends confirmation to the sampler with the time frame ID to **measure the time frame building time**.



Implementation within the ALFA framework, using FairMQ for transport.
See talks by:
Mohammad Al-Turany (Tue 14.4, 15:30)
Florian Uhlig (Tue 14.4, 15:15)

Traffic Shaping to relax the network requirements



Staggering of the transfers on the FLPs

Approach: Delay transfers of some FLPs by an offset, storing pending messages in a buffer.

delay = latency(sizeof(sub-time frame)) * priority

(measured latency)

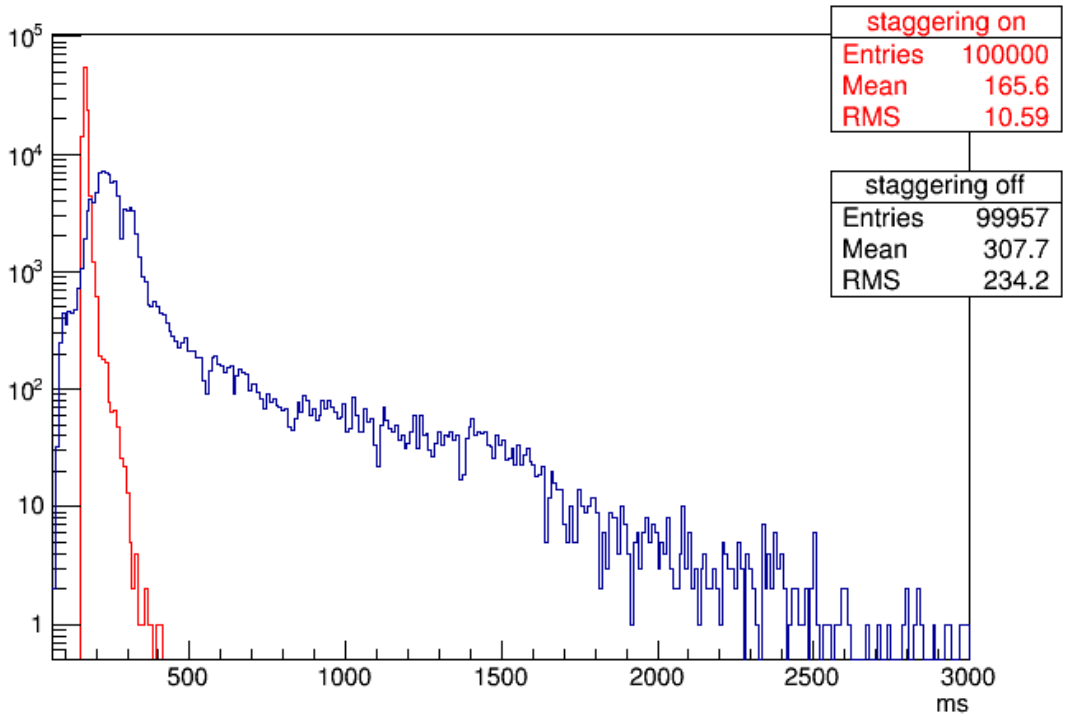
- The delay value can be further tuned if necessary.
- Priority is **configurable** per FLPSender process.
- Priority of 0 will disable the staggering on the FLP.
- Simultaneous transfers by several processes can be achieved by giving them same priority value.

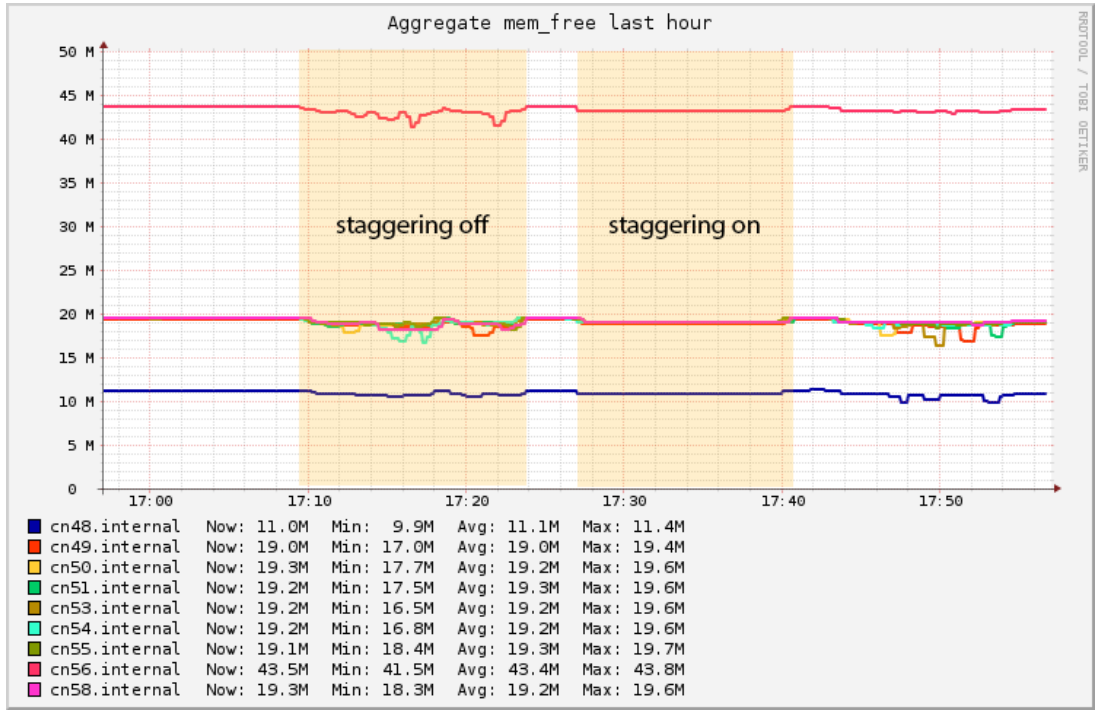
This approach requires more buffering (memory) on some FLPs, but relaxes the network load.

Effect of the traffic shaping

Distribution of the time frame building times.

Measured on the SyncSampler with a setup of 1 x 9 FLP x 9 EPN nodes.
2 FLP processes x 3 EPN processes per node. Total throughput of 15.3 GB/s (per node: 1.7 GB/s).
Message size (sub-time frame) for the tests: 17 MB.
Test systems: Intel Xeon E5520, AMD Opteron 6172; 16-48 GB RAM; QDR Infiniband (40 Gbps)



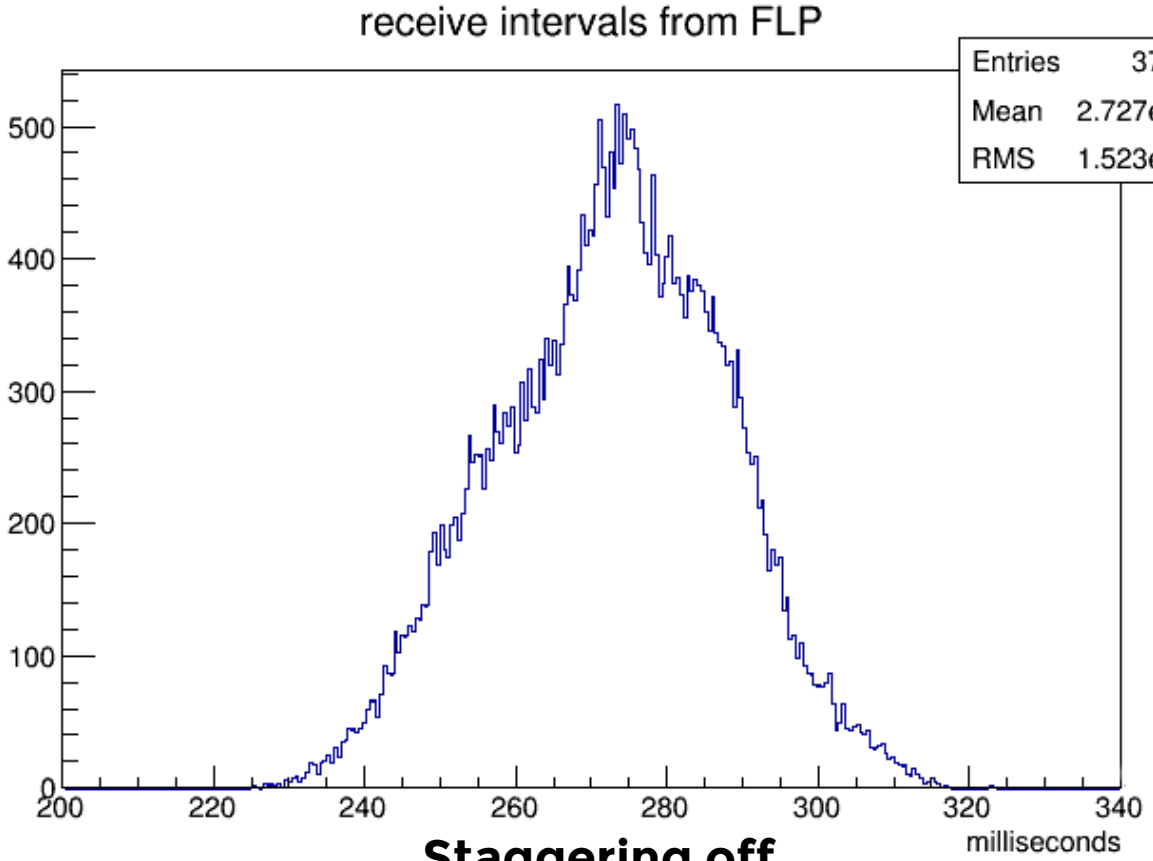


Ordered arrival compensates for the introduced delays for the staggering.

➡ Predictable behavior, stable memory usage, better performance.

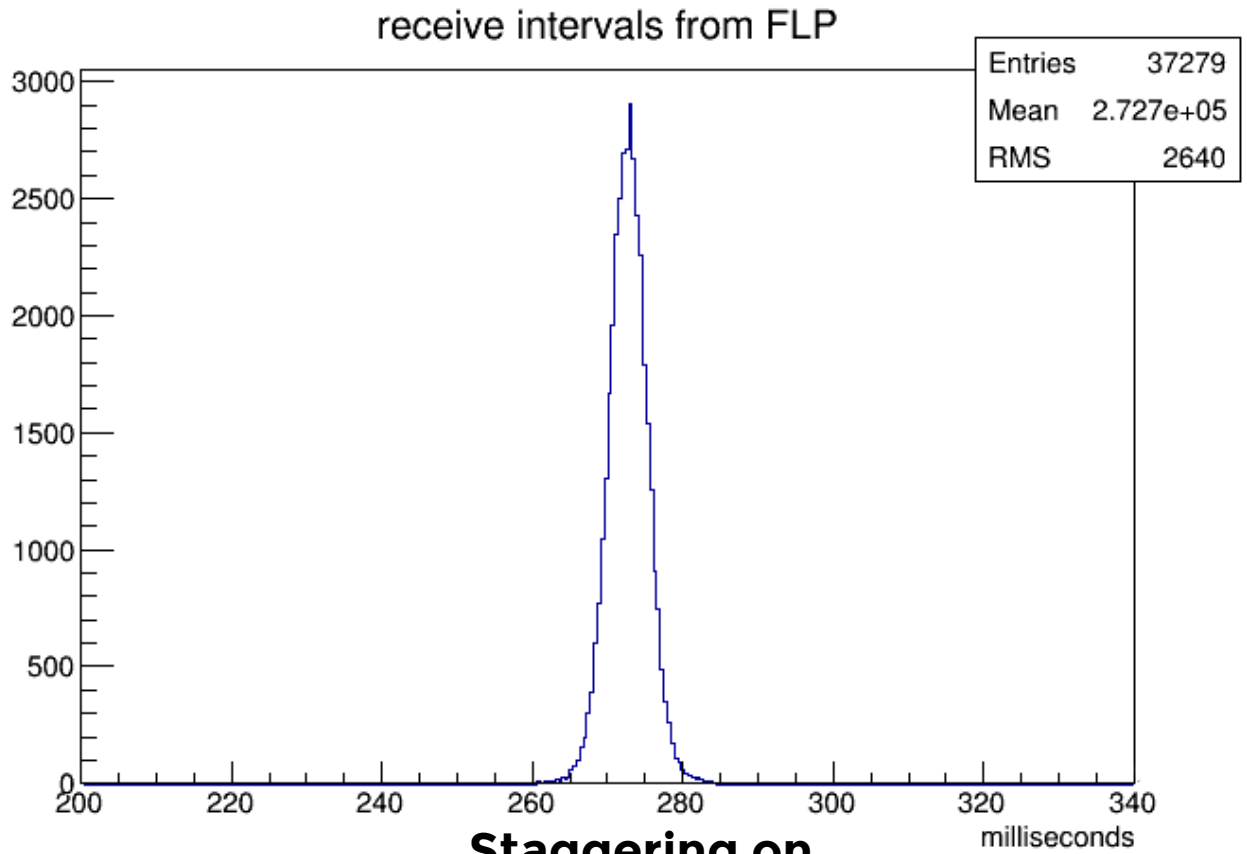
Effect of the traffic shaping (receiving pattern)

Intervals on an EPN between receiving from the same FLP.



Staggering off

RMS: 15230



Staggering on

RMS: 2640

With the traffic shaping, the traffic pattern is more predictable.

ALICE

Summary

- Traffic Shaping has been implemented on the FLPs to optimize the network & memory usage when multiple FLPs transfer data to the same EPN.
- By staggering the transfers from FLPs, simultaneous transfers to the same EPN can be avoided.
- The traffic shaping results in a more balanced and predictable network usage.
- The parameters for the traffic shaping are kept configurable, to allow any staggering pattern, tune it or to turn it off entirely.