A design study for the upgraded ALICE O^2 computing facility

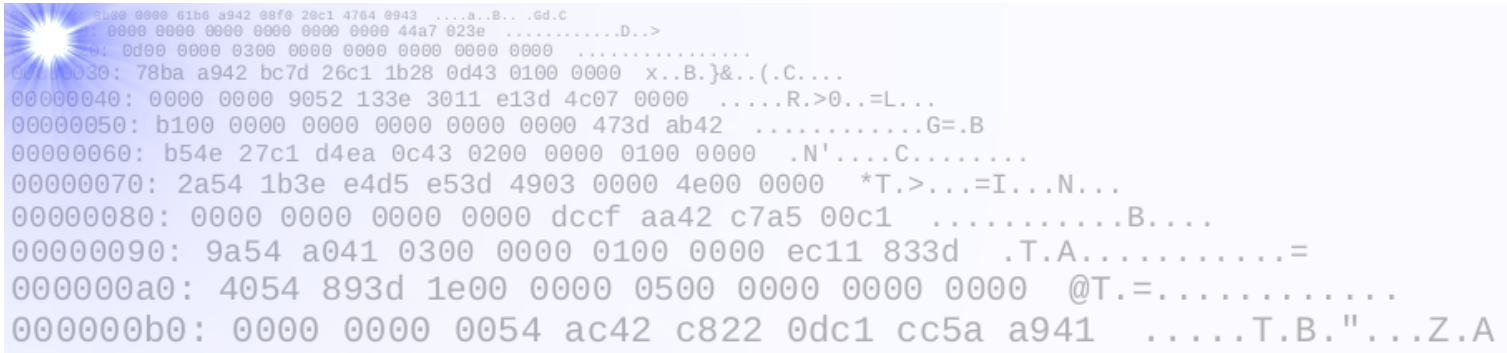
Matthias Richter
for the ALICE collaboration

Department of Physics, University of Oslo
CERN - European Organization for Nuclear Research

21st International Conference on Computing in High Energy and Nuclear Physics **CHEP2015** Okinawa Japan: April 13 - 17, 2015

Outline

- ALICE Upgrade and the O^2 Project
- Data Model for the upgraded ALICE computing facility
- Prototype development
- Performance
- Summary



Moving towards continuous online reconstruction of particle properties from raw data stream of 1 TByte/s

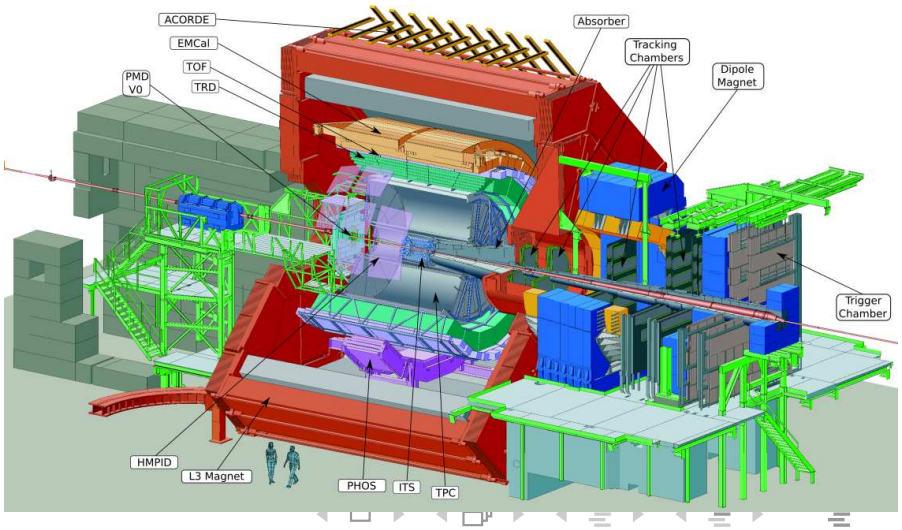
ALICE Upgrade and the O^2 Project

Targeting the time after LS2 upgrade of the Large Hadron Collider (LHC)

- ALICE letter of intent ▷ CERN-LHCC-2012-012 / LHCC-I-022
  Physics program requires to enhance vertexing capabilities in the low momentum range and substantial increase of data taking rate
- ALICE O^2 Project ▷ A novel combined online and offline computing system
  Computing requirements/challenges will be met by a combined online and offline facility

ALICE upgrade in (some) numbers

- LHC Pb-Pb luminosity $\mathcal{L} = 6 \cdot 10^{27} cm^{-2} s^{-1}$ interaction rate of **50 kHz**
- Time Projection Chamber (TPC) has an active detector volume of $5 \cdot 10^8$ "pixels"; collection time for one full volume $\sim 100 \mu s$
- Average event size **23 MByte** in min bias Pb-Pb data taking



Requirements and Constraints

LHC Pb-Pb luminosity $\mathcal{L} = 6 \cdot 10^{27} cm^{-2} s^{-1}$
→ **corresponds to interaction rate of 50 kHz**

Low signal to background ratio for the physics phenomena under investigation
→ **inspection of full data stream**

Min bias data taking, avrg event size 23 MByte
→ **input data rate >1 TByte/s**

Detectors with long data collection time
→ **continuous readout**

ALICE O^2 facility design

data compression based on reconstructed particle tracks
→ **online calibration**

Global reconstruction requires information of the full ALICE detector as input
→ **aggregation of corresponding data parts on processing nodes**

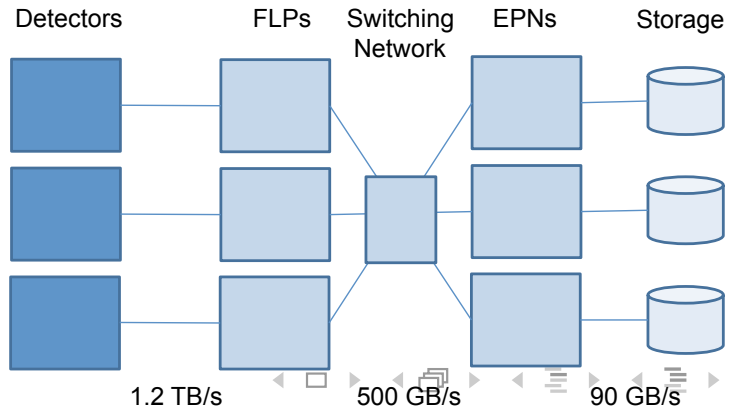
Multiple and heterogeneous detector systems
→ **flexible and modular framework**

Data Model and Data Flow

Components of the Data Flow

- **Detector Electronics:** Detector Front-end electronics (FEE) send data according to time intervals and/or trigger
- **First Level Processor (FLP):** Local aggregation of data in Timeframes (TF), the Timeframe data granularity is defined by heartbeat trigger events (HBE); Local reconstruction and calibration
- **Switching Network:** Physical transportation of timeframe data
- **Event Processing Node (EPN):** Aggregation of all sub-timeframes and global reconstruction of the full TF data sample
- **Storage:** Intermediate/permanent storage of pre-processed and compressed data

- Input: 8100 optical links
- 250 FLPs
- 1500 EPNs



schematic illustration

Prototype Development Strategy

What do we want to learn from a prototype?

- Is the data processing strategy feasible?

What should the prototype include?

- Framework
- Data Transportation
- Realistic Processing
- Process Distribution and Deployment

What do we want to achieve during the prototype development?

- a small scale but yet realistic processing topology

Prototype Development Assumptions

1. 92.5% of the data is generated by the TPC
   $\rightarrow$ focus on TPC processing
2. The data from the TPC front-end will arrive via multiple links in the FLP nodes
   $\rightarrow$ use present readout layout with 216 links
3. Local cluster reconstruction is running on hardware accelerator cards in real-time on the input streams
   $\rightarrow$ O^2 facility processing starts with clusters (space points) in the main memory of FLP nodes

Build upon ...

- Extensive experience with existing online processing system
  ALICE HLT: hardware, software framework, algorithms, ...
- Multi-process parallelization with the help of message queues

Software Framework

Alice O^2 software framework builds on

- ▷ ALFA software project (common ALICE and FAIR software development)
- ▷ FairMQ (FAIR messaging framework), currently using ZeroMQ

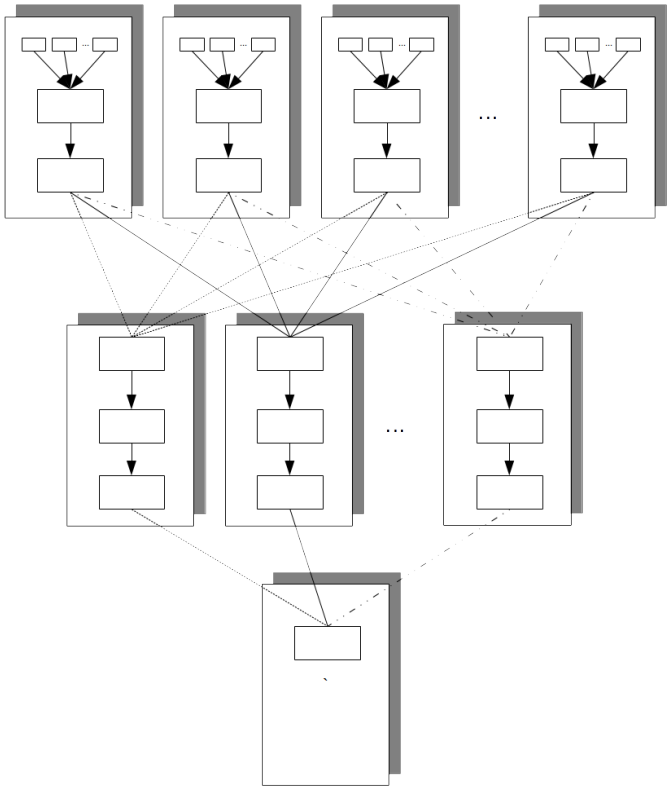
- A processing entity in Alice O^2 is called a *device*
- Devices implement
  - algorithms for detector reconstruction, calibration and global reconstruction
  - data transport functionality

Parallel processing strategy

- Processing of full data stream is distributed over multiple devices on multiple nodes
  $\Rightarrow$ multi-process
- A device and the implemented algorithm make use of threads
  $\Rightarrow$ multi-threading

See also at this conference: M. Al-Turany *ALFA: The new ALICE-FAIR software framework* .
Track 2 14/4/2015 15:30

Prototype processing topology: TPC online reconstruction



Implementing TPC reconstruction

- 216 Data Publishers on 36 nodes (one for each TPC slice)
- 36 FLPsender devices
  → 28 EPNreceiver devices
- 28 tracking devices
  → 28 track merger devices
- 1 data sink

Small scale test environment using existing infrastructure with ∼40 nodes

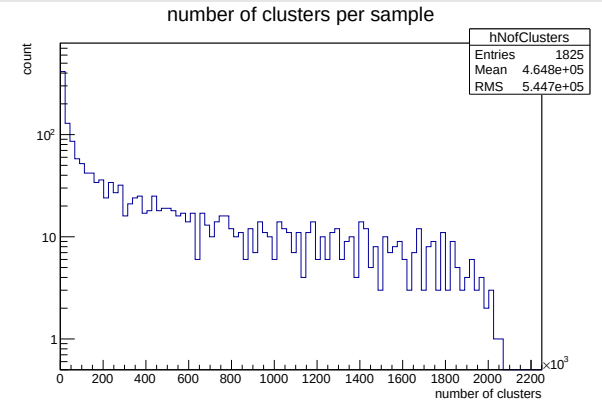
- ▷ 16 core Intel® Xeon® 2.26 GHz/ 24 core AMD Opteron™ 2.1 GHz
- ▷ GPU used as accelerator card for particle track finding

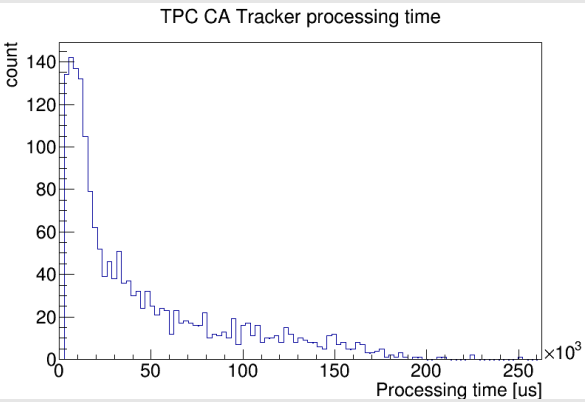
Processing Topology Performance

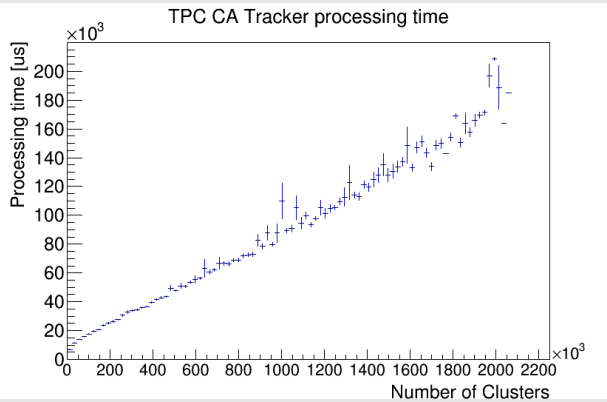
Key figures

- TPC reconstruction of individual events
- Timeframe data extrapolation
- Data transport performance

TPC Cellular Automaton tracker (track finder) on GPU nodes





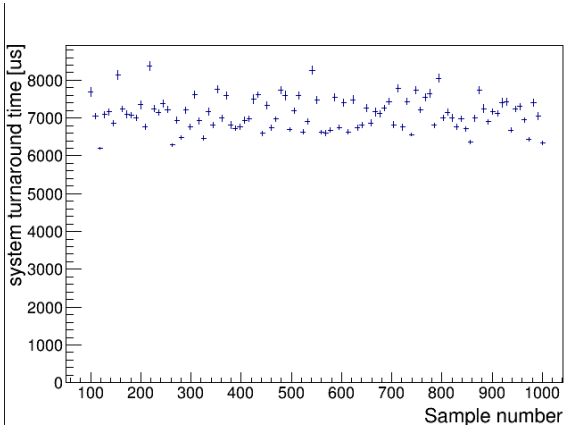


- distribution of number of clusters reflects the centrality distribution of min bias data sample
- TPC CA tracker processing time linear in number of clusters

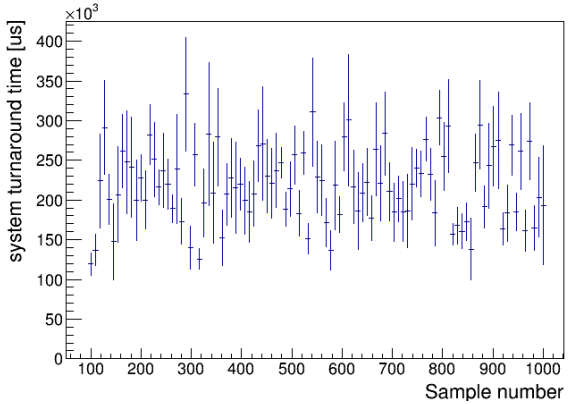
Reconstruction Performance of individual events

- Publishing data at ∼100Hz, average sample size 16 MByte
- Topology is processing aggregated size of 1.6 GByte/s

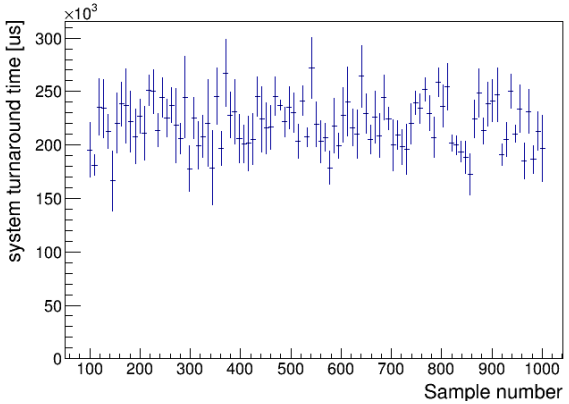
Cluster Publishers

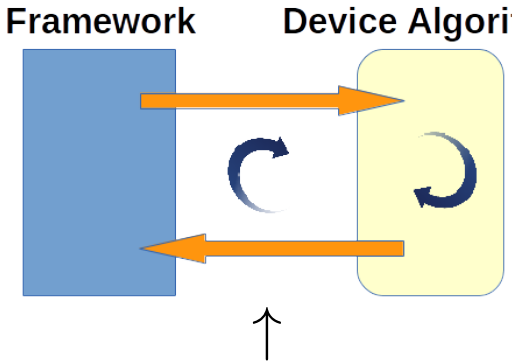


Tracker



Merger





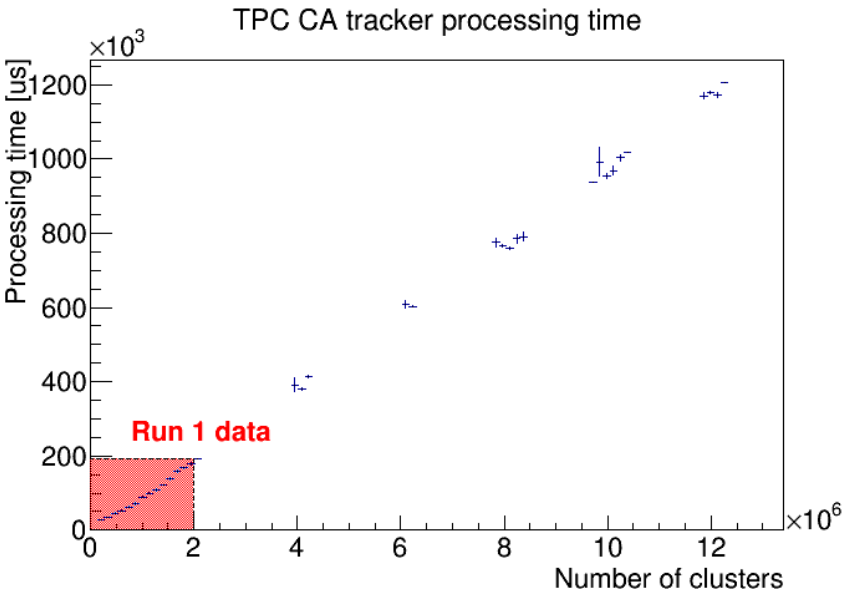
System turnaround time = time elapsed until next data sample is provided by framework

- at publishers: ∼7 ms
- at tracker/merger: ∼220 ms (28 EPN branches)

⇒ No saturation

Performance with extrapolated Timeframe Data

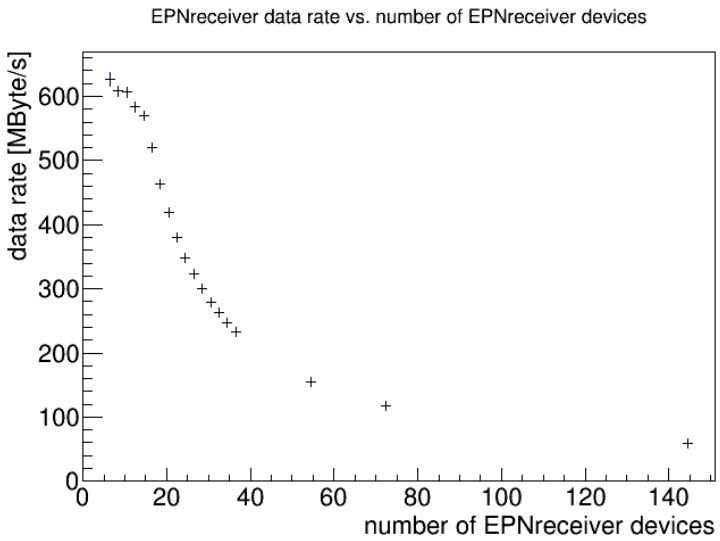
- Timeframe-like data has been produced by overlaying clusters of real Run 1 Pb-Pb events
- Clusters of individual events are shifted in z (drift direction) by constant offset
  - ▷ no realistic detector data as distortion effects are not considered
  - ▷ realistic data in terms of Timeframe size



- Linear trend in processing time continues
- Limitations for the event size scaling: internal memory of GPU
- Next step: extension of the CA Tracker device to provide cluster data of sliding window to GPU

Data Transportation

- Data transportation ships sub-timeframes from FLPs to EPNs
- Aggregates all parts belonging to a particular time frame on one Event Processing Node
- Implemented as devices in the FairMQ framework



- Test with fixed total input rate to the system
- Network protocol IP over Infiniband
- 16 core Intel® Xeon® 2.26 GHz/ 24 core AMD Opteron™ 2.1 GHz
- Data rate on the EPN decreases with increasing number of EPNreceiver devices in the configuration

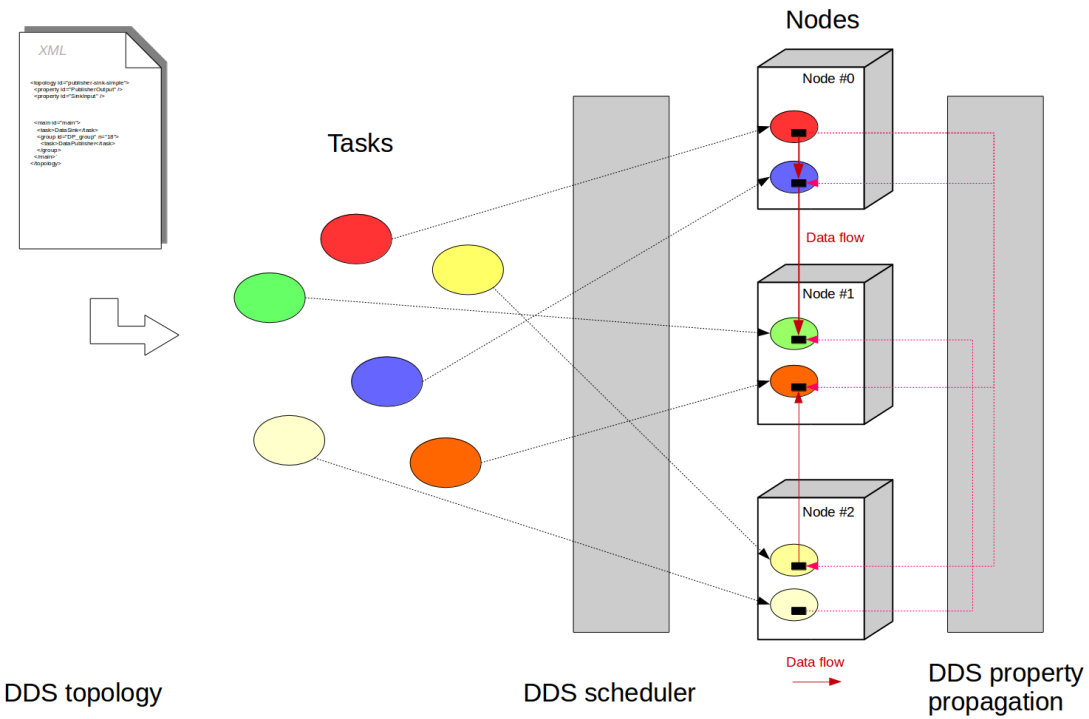
▷ Sustained data rate of up to 550 MByte/s, limited by the CPU consumption of the EPNreceiver device

see also at this conference: Track 1 14/4/2015 16:45 — A. Rybalchenko *Efficient time frame building for online data reconstruction in ALICE experiment*

Process Scheduling and Deployment

Dynamic Deployment System (DDS)

- Development within the ALFA project
- Translates a topology description into task schedule table
- Deploys tasks on processing nodes
- Property propagation among tasks



- XML configuration file defines tasks, collections and groups
- DDS Scheduler deploys and controls tasks on nodes
- Devices exchange properties via DDS
- Devices exchange data directly via messaging system

Summary and Outlook

- Emphasis on getting a realistic processing topology running in the Alice O^2 prototype
- FairMQ Framework is suited for the implementation of the distributed processing topology
- Sustained data aggregation rate up to 550 MByte/s per EPNreceiver, $\sim$ 10 GByte/s in the test system
- FLP to EPN data transportation prove to fulfill the requirement
- Efficient process scheduling and deployment system tested with the prototype
- System is ready for larger test on dedicated hardware