ALFA:
Next generation concurrent framework for ALICE and FAIR experiments

Mohammad Al-Turany
GSI-ExpSys/CERN-PH

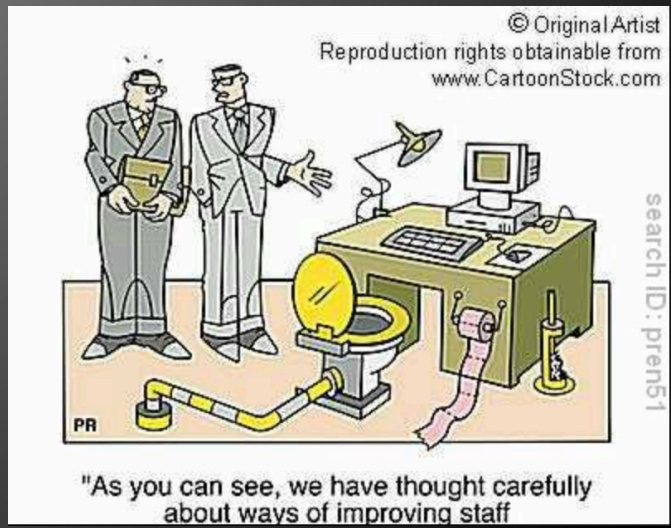
This talk

- Motivation: Why a new Framework?
- Basic features and components of ALFA
- Prototype for ALICE upgrade

FairRoot: a success

- Used for simulations and design studies for FAIR and Non-FAIR experiments
- It enhanced the synergy between the different groups
- Many useful tools where developed within FairRoot

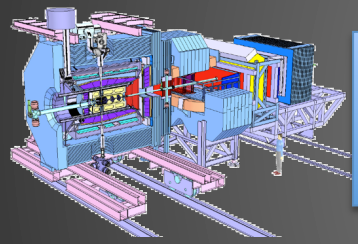
See Florian talk:
https://indico.cern.ch/event/349459/session/1/contribution/46



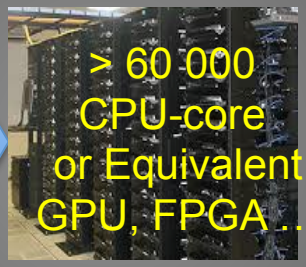
But: What about..

- Online computing?
  - Handling 1 TByte/s data transport in the online systems

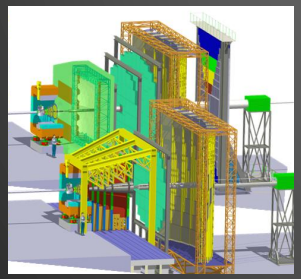
PANDA



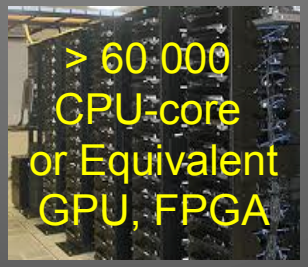
300 GB/s
20M Evt/s



< 1 GB/s



1 TB/s



1 GB/s

CBM

What about..

- Support heterogeneous architectures
  - Accelerator cards (GPUs, Xeon Phi)
- Concurrency?
  - Multi-/Many-Core
  - SIMD

ALICE LS2 Upgrade - Strategy

ALICE

More than 1 TByte/s detector readout

- Storage bandwidth limited to ~20 GByte/s (design decision/cost)
- Many physics probes have low S/B:
  classical trigger/event filter approach not efficient

⇨ **Store only reconstruction results, discard raw data**

- Data reduction by (partial) online reconstruction and compression
- >100.000 cores + GPUs + FPGAs

⇨ **Implies much tighter coupling between online and offline reconstruction software**

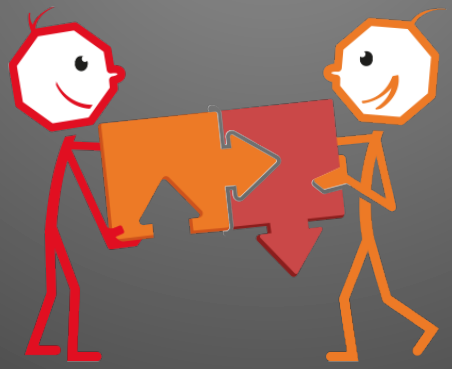
ALICE and FAIR: Why?

Two projects – same requirements

Massive data volume reduction (1 TByte/s input)

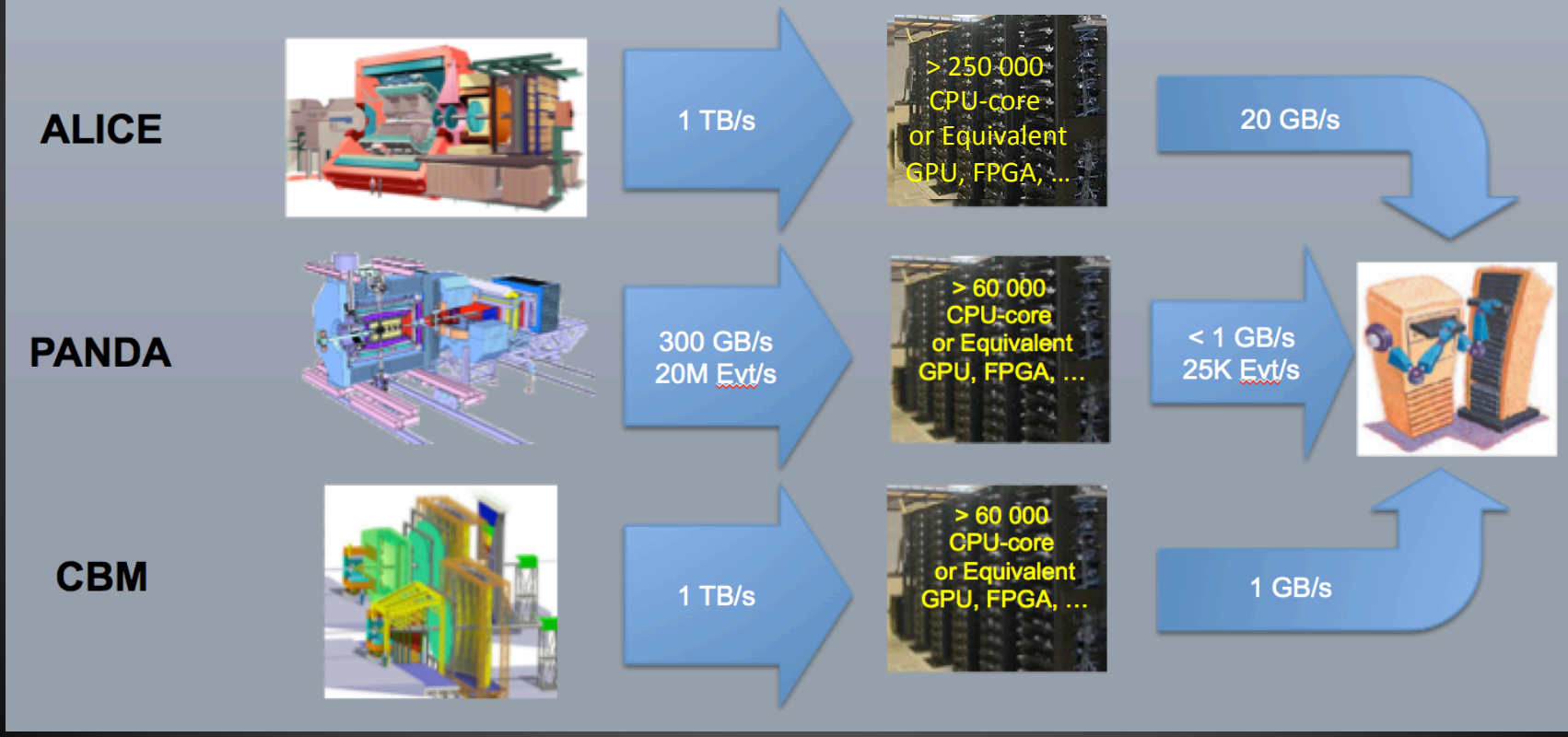
- Data reduction by (partial) online reconstruction
- Online reconstruction and event selection

Much tighter coupling between online and offline reconstruction software



Lets work together

How to distribute the processes?
How to manage the data flow?
How to recover processes when they crash?
How to monitor the whole system?
……



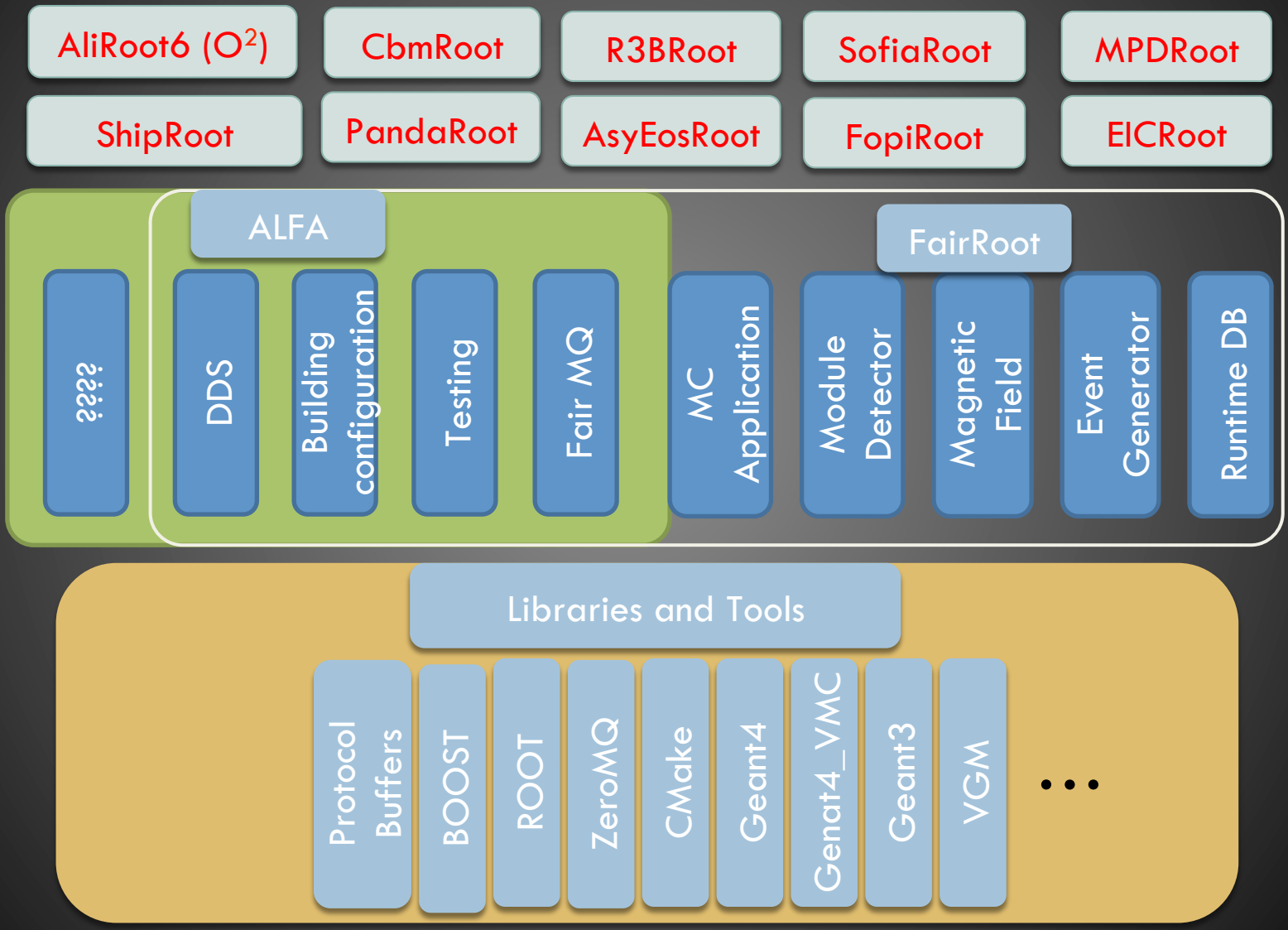
Common Strategy

- Massive data volume reduction
  - Data reduction by (partial) online reconstruction and compression
- Much tighter coupling between online and offline reconstruction software

ALFA

- A modular set of packages that contains:
  - Transport layer (FairMQ, based on: ZeroMQ, nanomsg)
  - Configuration tools
  - Management and monitoring tools
- A data-flow based model (Message Queues based multi-processing).
- Provide unified access to configuration parameters and databases.

ALFA and FairRoot



Correct balance between reliability and performance

- Multi-process concept with message queues for data exchange
  - Each "Task" is a separate process, which:
    - Can be multithreaded, SIMDized, …etc.
    - Can run on different hardware (CPU, GPU, XeonPhi, …etc.)
    - Be written in an any supported language (Bindings for 30+ languages)
  - Different topologies of tasks can be adapted to the problem itself, and the hardware capabilities.

Scalability through multi-processing with message queues?

Each process assumes limited communication and reliance on other processes.

- No locking, each process runs with full speed
- Easier to scale horizontally to meet computing and throughput demands (starting new instances) than applications that exclusively rely on multiple threads which can only scale vertically.

ALFA uses FairMQ to connect different pieces together

FairMQ

ØMQ

nanomsg

Heterogeneous Platforms: Message format

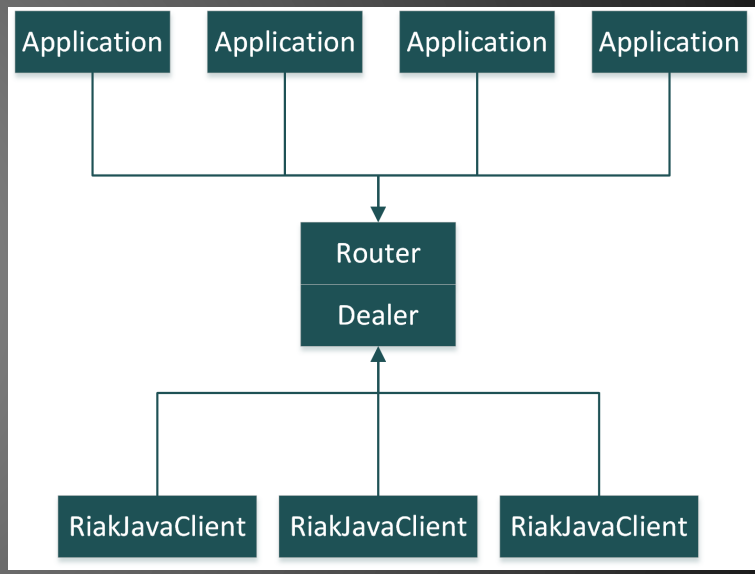
- The framework does not impose any format on messages.
- It supports different serialization standards
  - BOOST C++ serialization
  - Google's protocol buffers
  - ROOT
  - User defined

Tom Van Steenkiste

Parameter management

Distributed Model based on Riak

- high availability
- scalability
- fault tolerance
- configurable



How to deploy ALFA on a laptop, few PCs or a cluster?

- DDS: Dynamic Deployment System
  - Users describe desired tasks and their dependencies using topology files
  - The system takes so called “topology file” as the input.
  - Users are provided with a WEB GUI to create topology (Can be created manually as well).

DDS

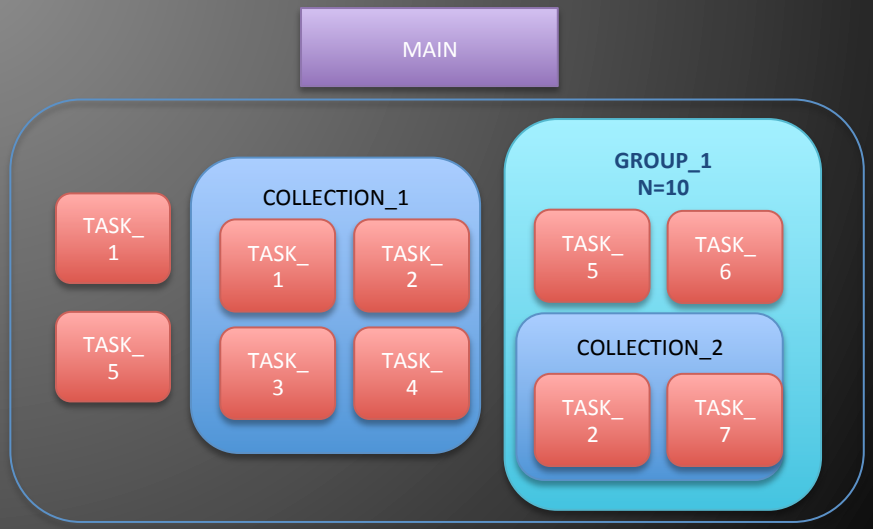
One of the key challenges of the FairMQ approach:
Process Management for 10.000 to 100.000 devices

- Control
- Monitoring
- Configuring

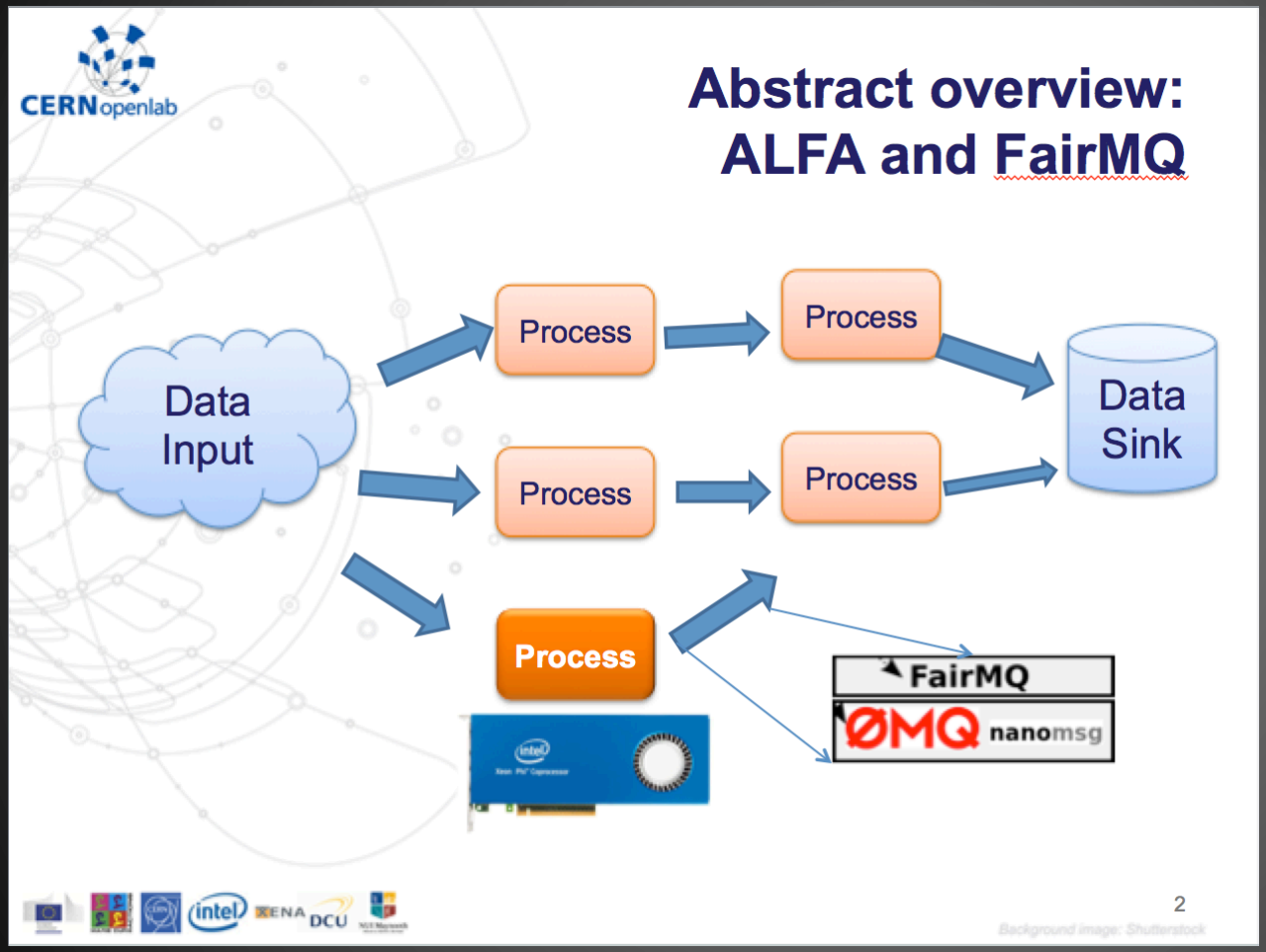
Dynamic Deployment System

- Separate module in FairRoot / ALFA
- Xml description of process topology

http://dds.gsi.de/



Xeon Phi



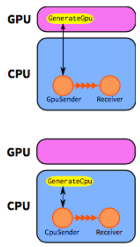
Aram SANTOGIDIS

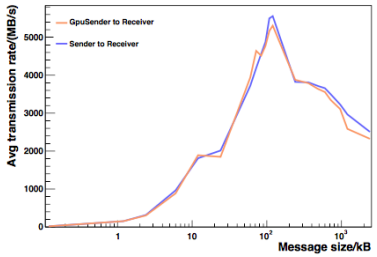
http://indico.cern.ch/event/304944/session/9/contribution/27

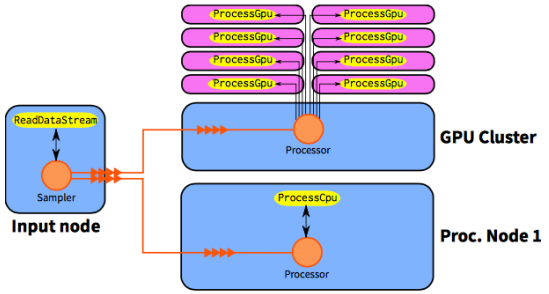
GPUs in ALFA

GPUs and Message Queues

- Explore communication/data transfer to GPUs
- FairMQ: implementation of Message Queues in the FairRoot framework (☞Apr 14: M. Al-Turany, A. Rybalchenko, F. Uhlig)
- Test system with implementation of Circle Hough algorithm
  - Modular structure
  - CPU and GPU version of processing task
  - FairMQ: stream input data to CPU/GPU processing tasks
  - Maximum flexibility of architecture and data transfer interface







L. Bianchi | Online Tracking with GPUs at PANDA | CHEP2015

13 / 14

Mitglied der Helmholtz-Gemeinschaft

Ludovico BIANCHI

http://indico.cern.ch/event/304944/session/1/contribution/363

Prototype for Alice O2

- Is the data processing strategy feasible?
- Can we create a small scale but yet realistic processing topology ?

The prototype:

- In ALICE 92.5% of the data is generated by the TPC

  focus on TPC processing

- The data from the TPC front-end will arrive via multiple links in the FLP nodes

  use present readout layout with 216 links

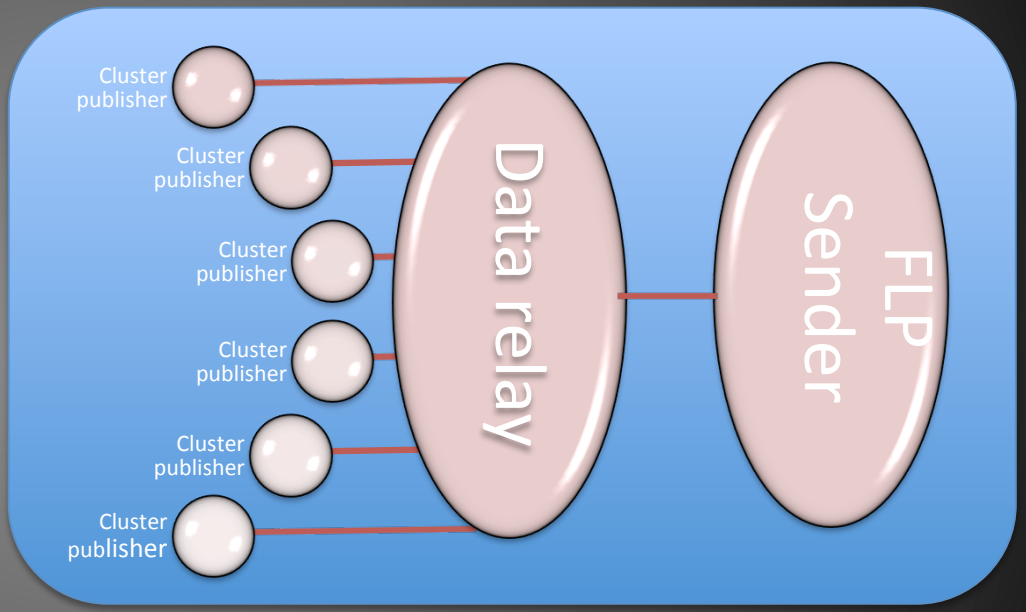
- Local cluster reconstruction is running on hardware accelerator cards in real-time on the input streams

Prototype start with clusters (space points) in the main memory of FLP nodes

Matthias Richter

Proto-Type: FLP devices

- 36 Data sources
  - 36 x 6 cluster publisher
  - 36 Merger (Data relay)
  - 36 FLP Sender



216+36+36 = 288 processes

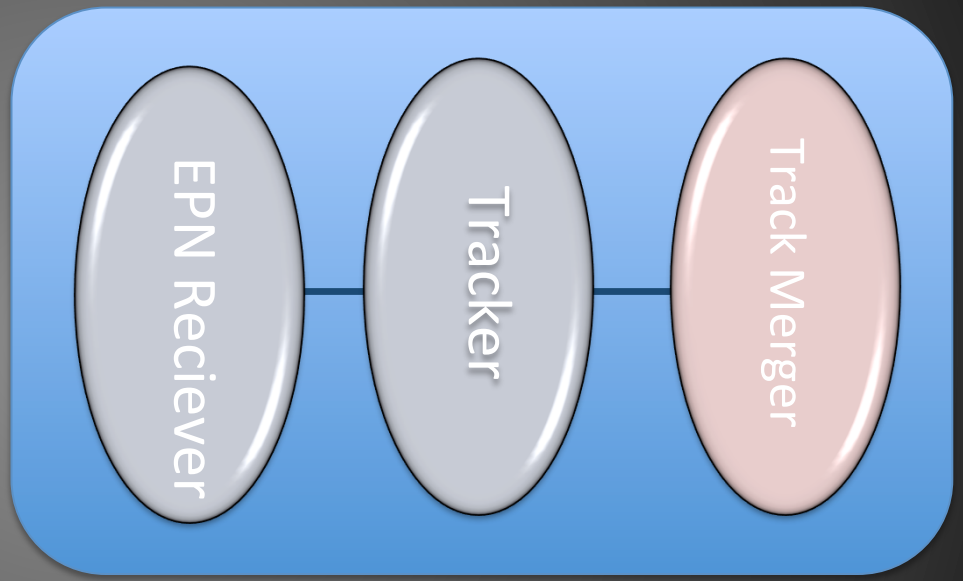
FLP: First level data processer

use present readout layout with 216 links

Matthias Richter

Proto-Type: EPN devices

- 28 Data consumers
  - 28 recievers
  - 28 Trackers (GPU)
  - 28 Track mergers

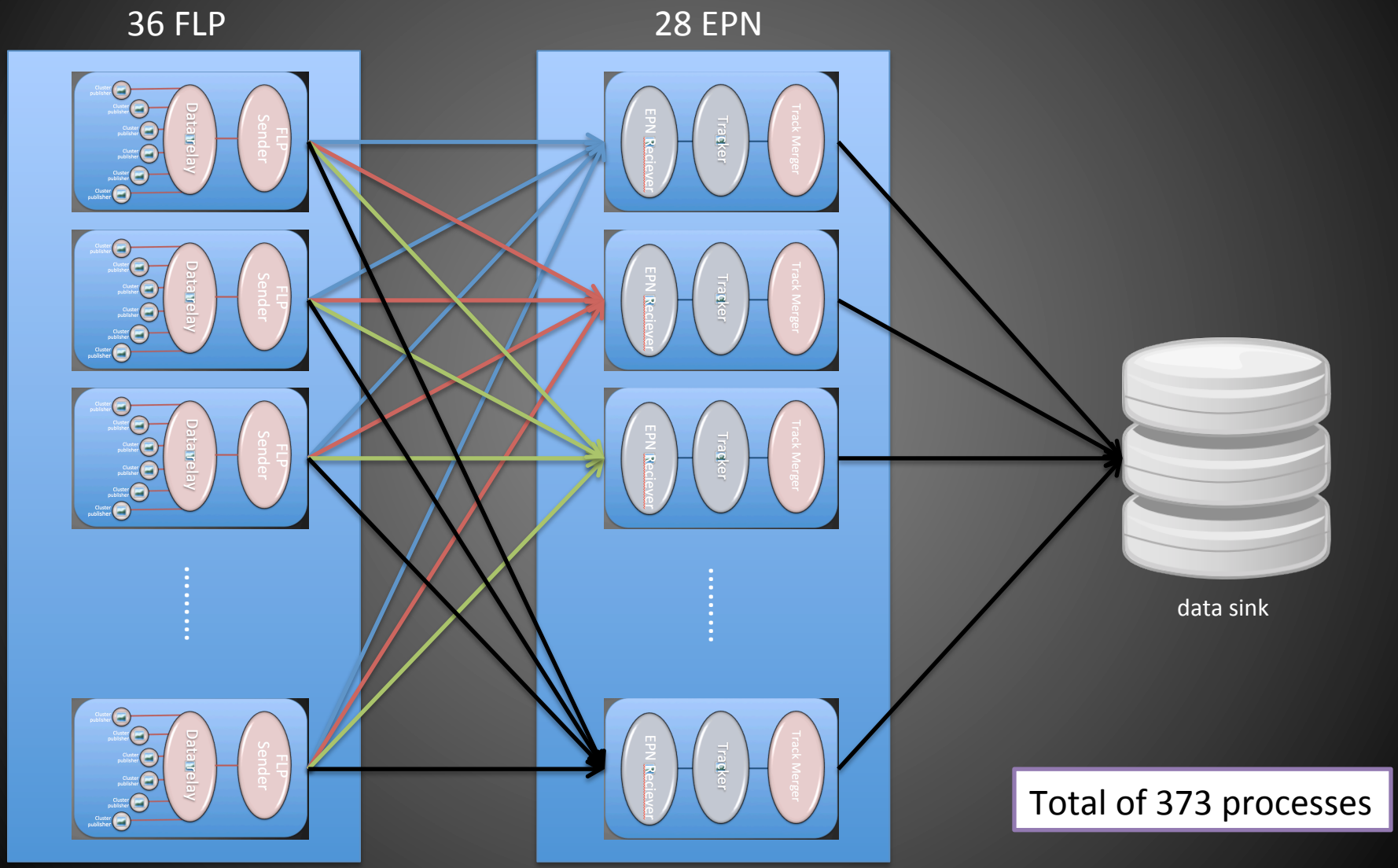


28+28+28 = 84 processes

EPN: Event Processing Node

Implementing TPC reconstruction

Matthias Richter



Total of 373 processes

Test Hardware

- Small scale test environment (40 nodes) using parts of existing ALICE HLT development cluster :
  - 16 core Intel Xeon 2.26 GHz
  - 24 core AMD Opteron 2.1 GHz
  - GPU used as accelerator card for particle track finding
- Network protocol IP over InfiniBand

Matthias Richter

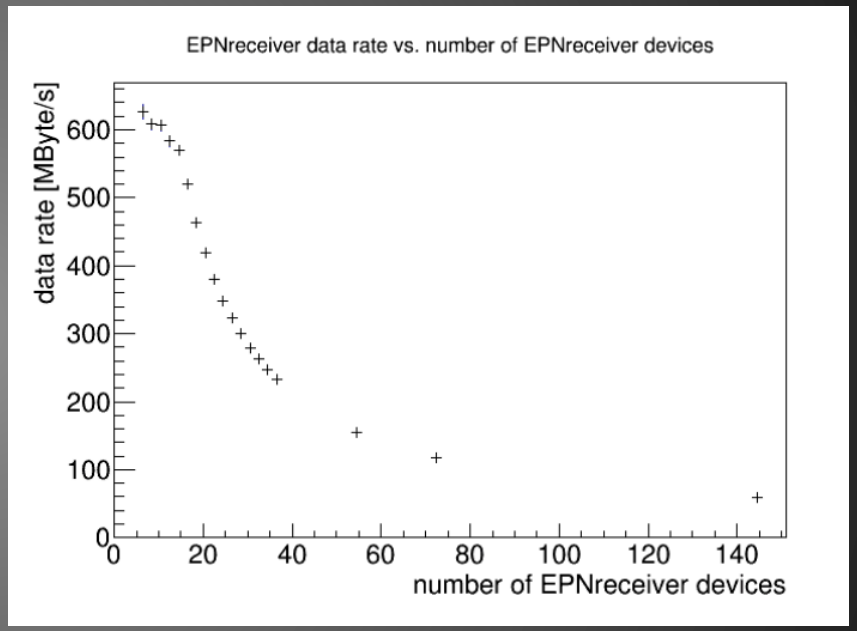
Results form Prototype:

- the topology is processing aggregated size of 1.6 GByte/s (limited by the cluster publishers)
- 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

Results form Prototype: EPN

Matthias Richter

- EPNreceiver sustained data aggregation rate up to about ***600 MByte/s*** per node (limited by the CPU consumption of the EPNreceiver device)
- Data rate on the EPN decreases with increasing number of EPNreceiver devices in the configuration



More technical details about the prototype can be found here:

Alexey RYBALCHENKO:
Efficient time frame building for online data reconstruction in ALICE experiment
https://indico.cern.ch/event/304944/session/1/contribution/353

Matthias RICHTER:
A design study for the upgraded ALICE O2 computing facility
https://indico.cern.ch/event/304944/session/1/contribution/439

Summary

- ALFA is under continuous development but already usable now
- Modular design allow us to replace, add or remove parts on the fly
- Test with Riak are very promising and it seems to fulfill the requirement for online/offline parameter DB
- DDS was used successfully to distribute tasks and propagate all needed properties for a system of about 10000 processes

backup

DDS

Connecting the FairMQ devices/tasks requires knowledge of connection parameters

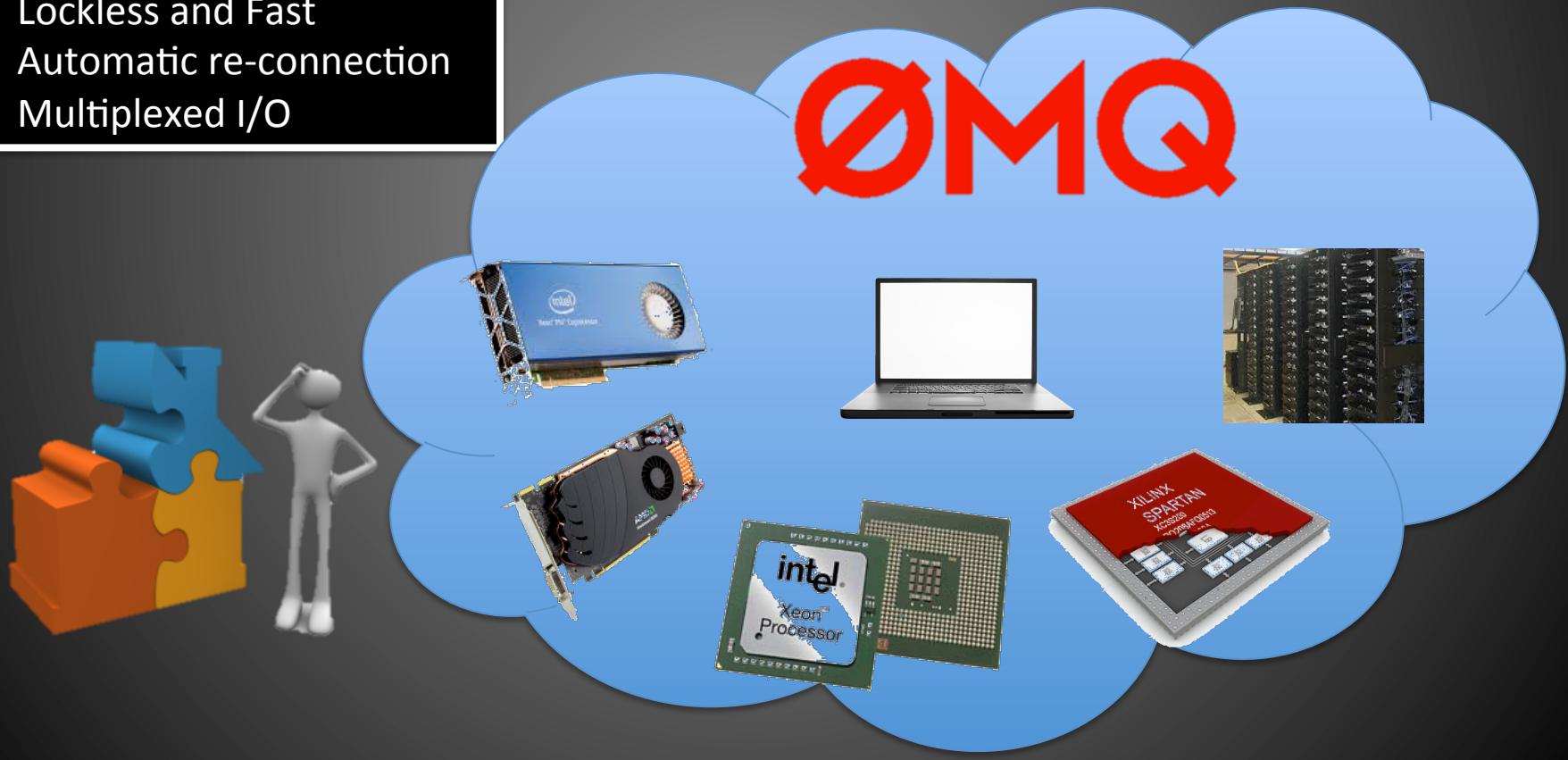
DDS supports dynamic configuration with key-value propagation

Devices (user tasks)	startup time*	propagated key-value properties
2721 (1360 FLP + 1360 EPN + 1 Sampler)	17 sec	~ $6x10^6$
5441 (2720 FLP + 2720 EPN + 1 Sampler)	58 sec	~ $23x10^6$
10081 (5040 FLP + 5040 EPN + 1 Sampler)	207 sec	~ $77x10^6$

* **startup time** - the time which took DDS to distribute user tasks, to propagate all needed properties, plus the time took devices to bind/connect and to enter into RUN state.

A cloud that let you connect different pieces together

- BSD sockets API
- Bindings for 30+ languages
- Lockless and Fast
- Automatic re-connection
- Multiplexed I/O



Another one is under development by the original author of ZeroMQ

nanomsg

- Pluggable Transports:
  - ZeroMQ has no formal API for adding new transports (Infiniband, WebSeockets, etc). nanomsg defines such API, which simplifies implementation of new transports.
- Zero-Copy:
  - Better zero-copy support with RDMA and shared memory, which will improve transfer rates for larger data for inter-process communication.
- Simpler interface:
  - simplifies some zeromq concepts and API, for example, it no longer needs Context class.
- Numerous other improvements, described here: http://nanomsg.org/documentation-zeromq.html
- FairRoot is independent from the transport library
  - Modular/Pluggable/Switchable transport libraries.

Running the Zero MQ performance test on the DAQ test cluster