25
NEC'2015
XXV International Symposium
on Nuclear Electronics & Computing
CERN

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

CBM collaboration 2003

How it started?

- We need simulations for the LOI
- It has to be easy, fast, reliable, ..etc
- We have no manpower for software
- We need it yesterday

FLUKA

ROOT

PAW

Geant4

VMC

Pythia

AliRoot

Geant3

Urqmd

Hydra

CbmRoot Framework

Lightweight Framework based on ROOT

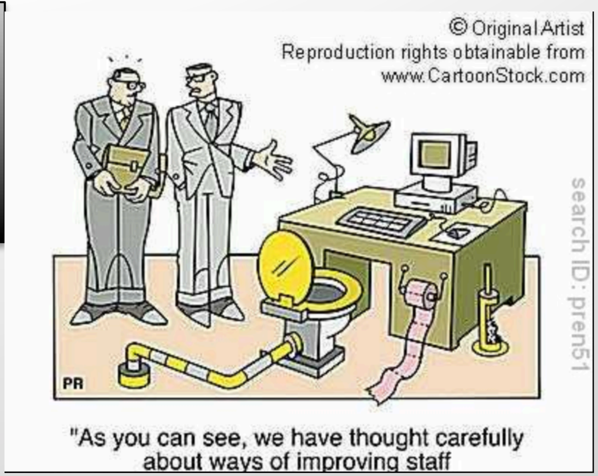
VMC and VGM for simulation

TGeoManager for Simulation and Reconstruction

Eve (Alice Event display) as base for a general event display

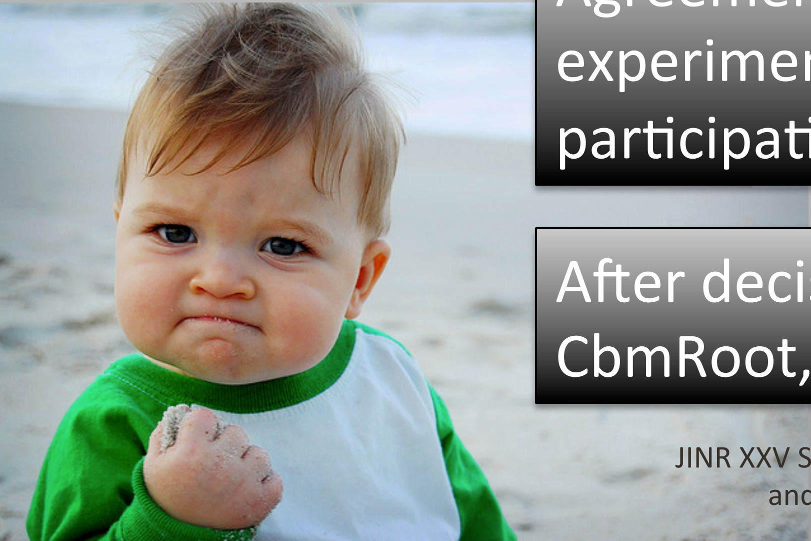
Hades oracle interface and run time database

Software for FAIR Experiments (FairRoot)



"As you can see, we have thought carefully about ways of improving staff

FAIR and non-FAIR experiments join the effort to build one platform for simulation and reconstruction software

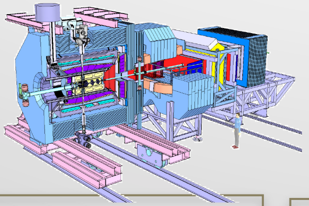


Agreement between GSI-IT management and the experiments to create a core team in the IT with participation of the experiment

After decision by Panda collaboration to use CbmRoot, the common part was called FairRoot

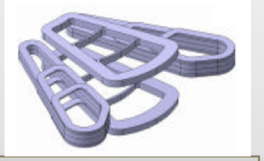
FairRoot

Start testing the VMC concept for CBM

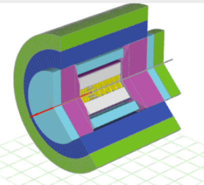


Panda decided to join->

FairRoot: same Base package for different experiments

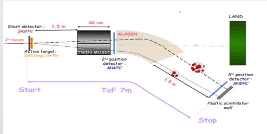


R3B joined

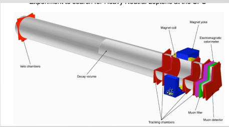


EIC (Electron Ion Collider BNL)

EICRoot



SOFIA (Studies On Fission with Aladin)



SHIP - Search for HIdden Particles

2004

2006

2010

2011

2012

2013

2014

First Release of CbmRoot

MPD (NICA) start also using FairRoot

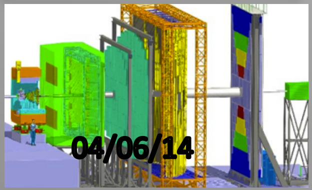
ASYEOS joined (ASYEOSRoot)

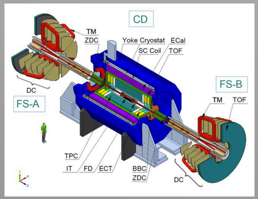
GEM-TPC seperated from PANDA branch (FOPIRoot)

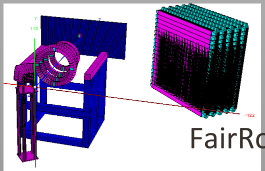
CALIFA (CALorimeter for the In Flight detection of γ rays and light charged pArticles)

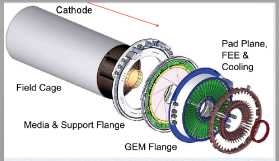
ENSAR-ROOT

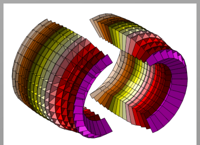
Collection of modules used by structural nuclear phsyics exp.



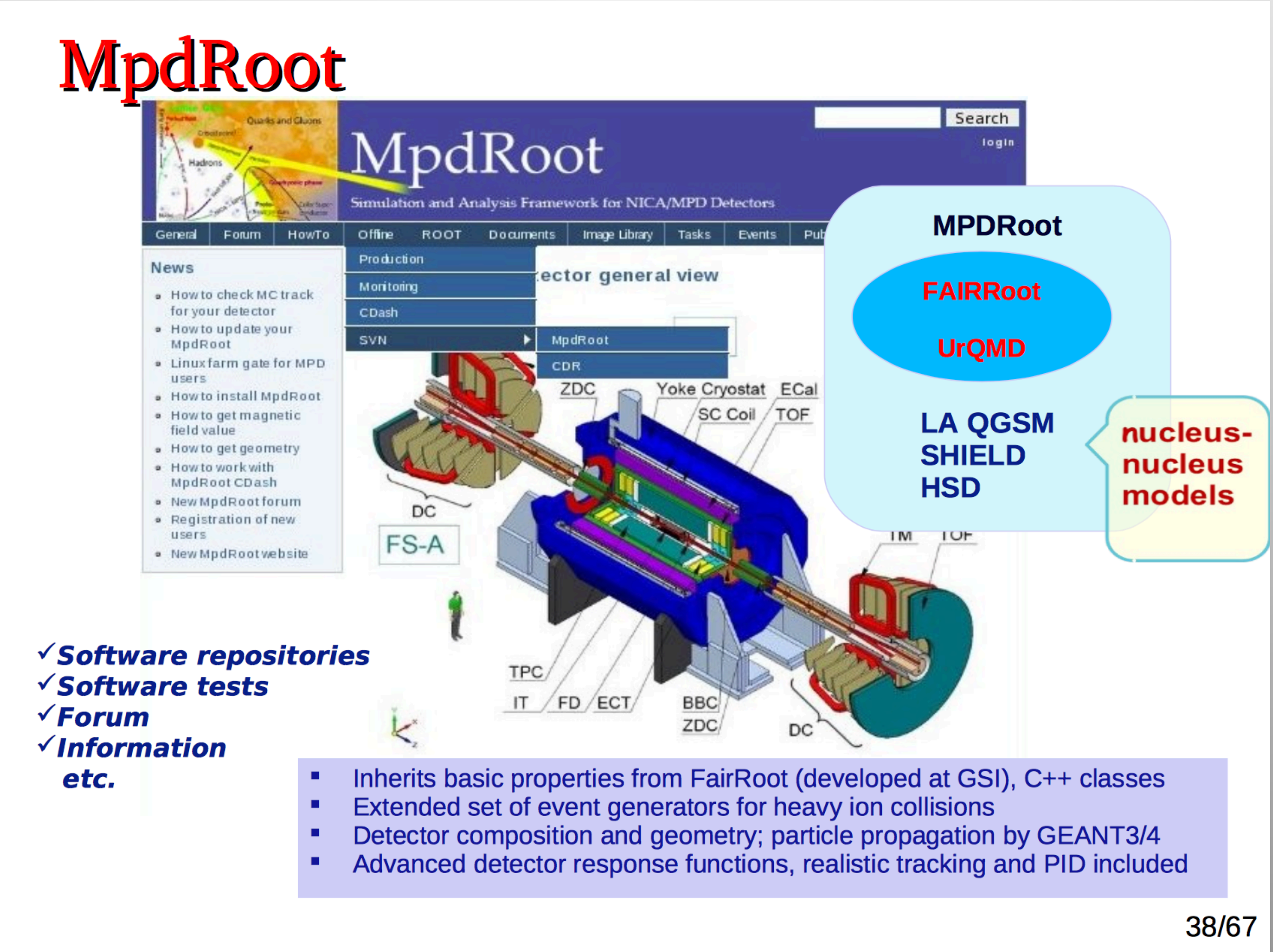








Used by the MPD at NICA since 2006



MpdRoot

MpdRoot

Simulation and Analysis Framework for NICA/MPD Detectors

News

- How to check MC track for your detector
- How to update your MpdRoot
- Linux farm gate for MPD users
- How to install MpdRoot
- How to get magnetic field value
- How to get geometry
- How to work with MpdRoot CDash
- New MpdRoot forum
- Registration of new users
- New MpdRoot website

✓*Software repositories*
✓*Software tests*
✓*Forum*
✓*Information*
etc.

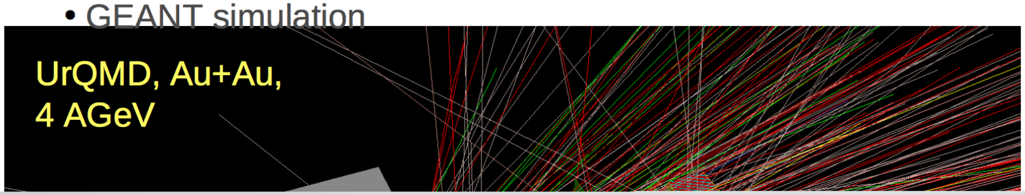
- Inherits basic properties from FairRoot (developed at GSI), C++ classes
- Extended set of event generators for heavy ion collisions
- Detector composition and geometry; particle propagation by GEANT3/4
- Advanced detector response functions, realistic tracking and PID included

38/67

BMNROOT software framework

- Detector geometry
- A+A event generators
- GEANT simulation

- Track reconstruction
- Particle identification
- Physics analysis



Software

Framework: BmnRoot – branch of FairRoot

Reconstruction:

several developments ongoing

the most advanced: Cellular Automaton track reconstruction method - adaptation of CBM so-called L1 tracking (following the synergy paradigm) and CBM STS detector digitization and hit finding scheme

It enhanced the synergy between the different groups

useful tools that where developed within FairRoot are available for all experiments

FairRoot is used for simulations and design studies by FAIR and other experiments

Are we done? What is next?

hell
no
gograph
gg53990663 www.gograph.com

??

What about

Heterogeneous architectures

- Accelerator cards (GPUs, Xeon Phi, etc)

Concurrency?

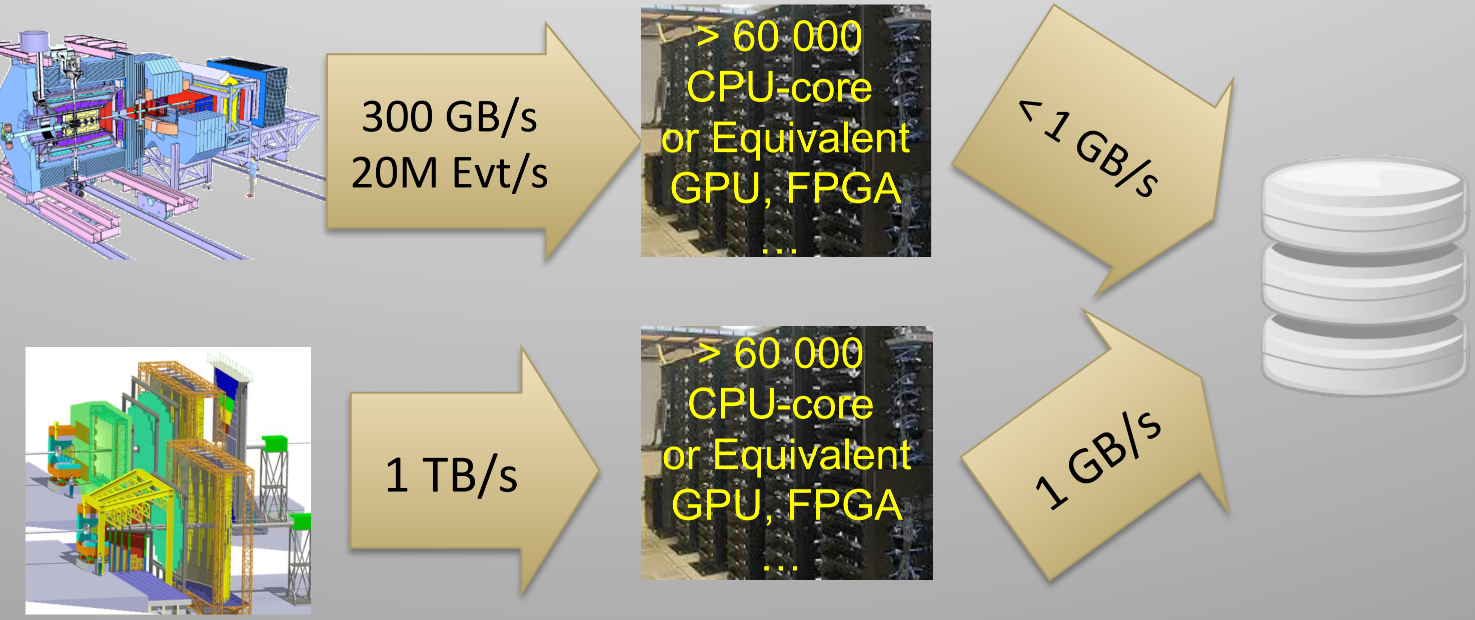
- Multi-/Many-Core
- SIMD

Online computing?

Handling 1 TByte/s data transport in the online systems

PANDA

CBM



300 GB/s
20M Evt/s

1 TB/s

> 60 000
CPU-core
or Equivalent
GPU, FPGA
…

> 60 000
CPU-core
or Equivalent
GPU, FPGA
…

< 1 GB/s

1 GB/s

ALICE LS2 Upgrade - Strategy

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

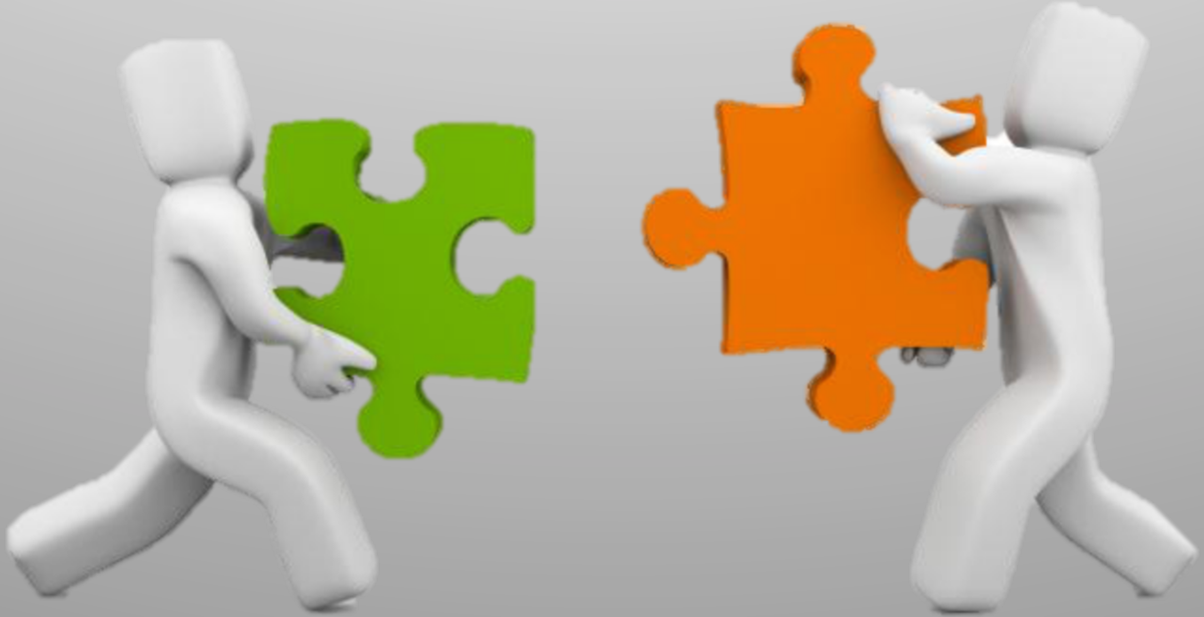
Implies much tighter coupling between online and offline reconstruction software

Two projects – same requirements

Massive data volume reduction (1 TByte/s input)

Data reduction by (partial) online reconstruction

Online reconstruction and event selection



ALFA

A modular set of packages that contains:

FairMQ

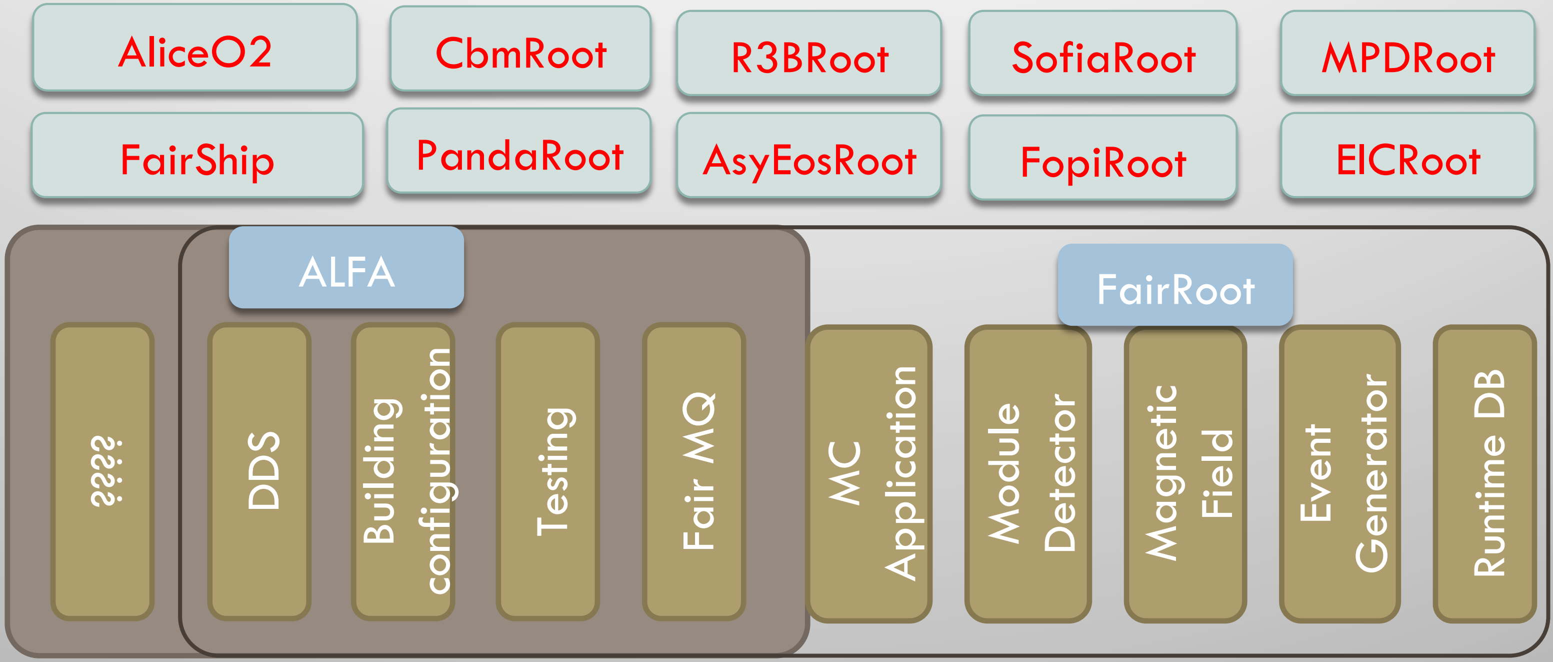
Configuration tools

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



AliceO2

CbmRoot

R3BRoot

SofiaRoot

MPDRoot

FairShip

PandaRoot

AsyEosRoot

FopiRoot

EICRoot

ALFA

????

DDS

Building configuration

Testing

Fair MQ

FairRoot

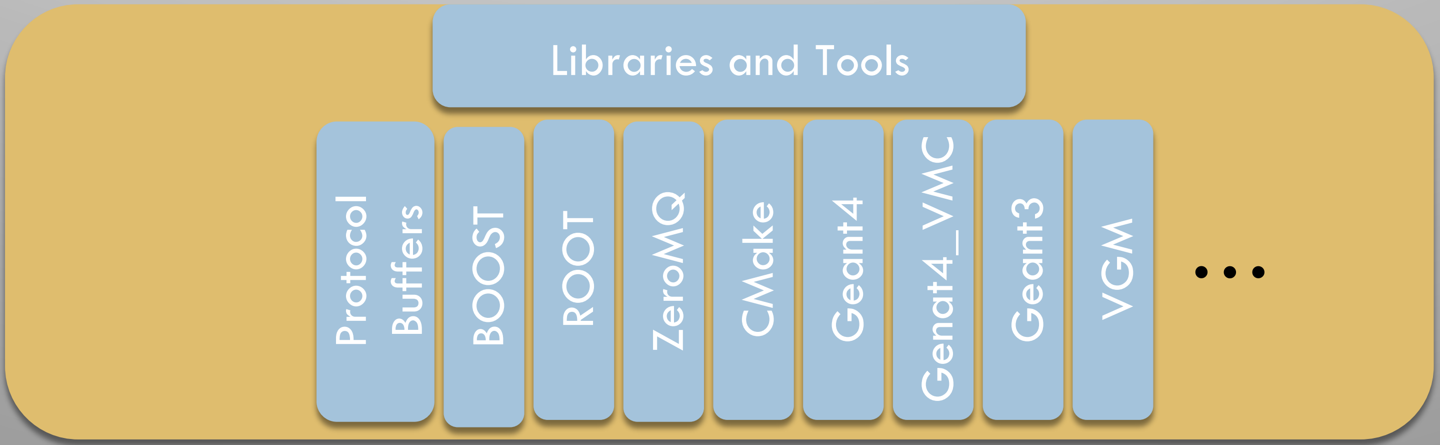
MC Application

Module Detector

Magnetic Field

Event Generator

Runtime DB



Libraries and Tools

Protocol Buffers

BOOST

ROOT

ZeroMQ

CMake

Geant4

Genat4_VMC

Geant3

VGM

…

Correct balance between reliability and performance

Each "Task" is a separate process, which:

- Can be multithreaded, SIMDized, …etc.
- runs on different hardware (CPU, GPU, …, 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

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

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

Users are provided with a WEB GUI to create topology (Can be created manually as well).

The system takes so called “topology file” as the input.

DDS

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

Control

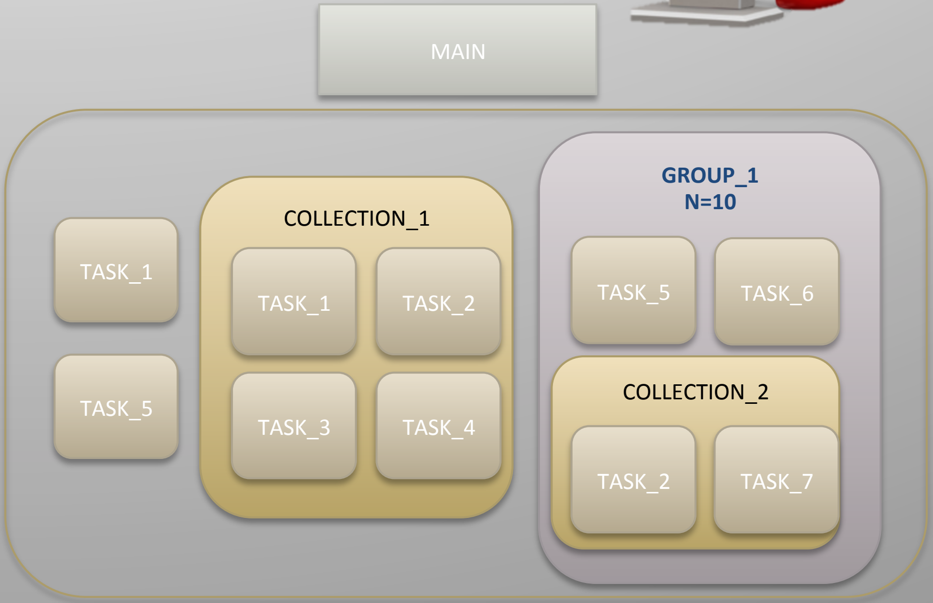
Monitoring

Configuring

Dynamic Deployment System

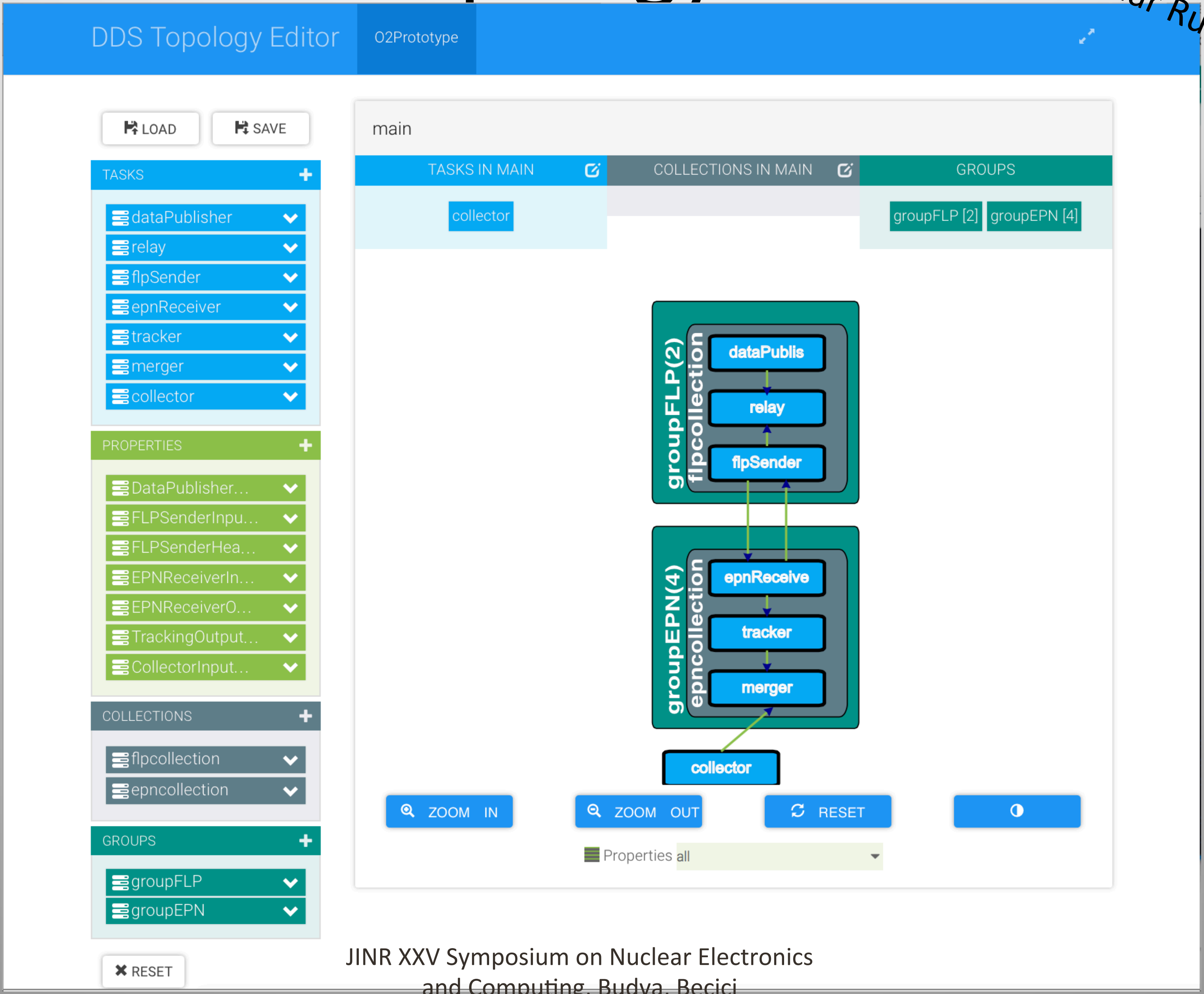
- Separate module in FairRoot / ALFA
- Xml description of process topology

http://dds.gsi.de/



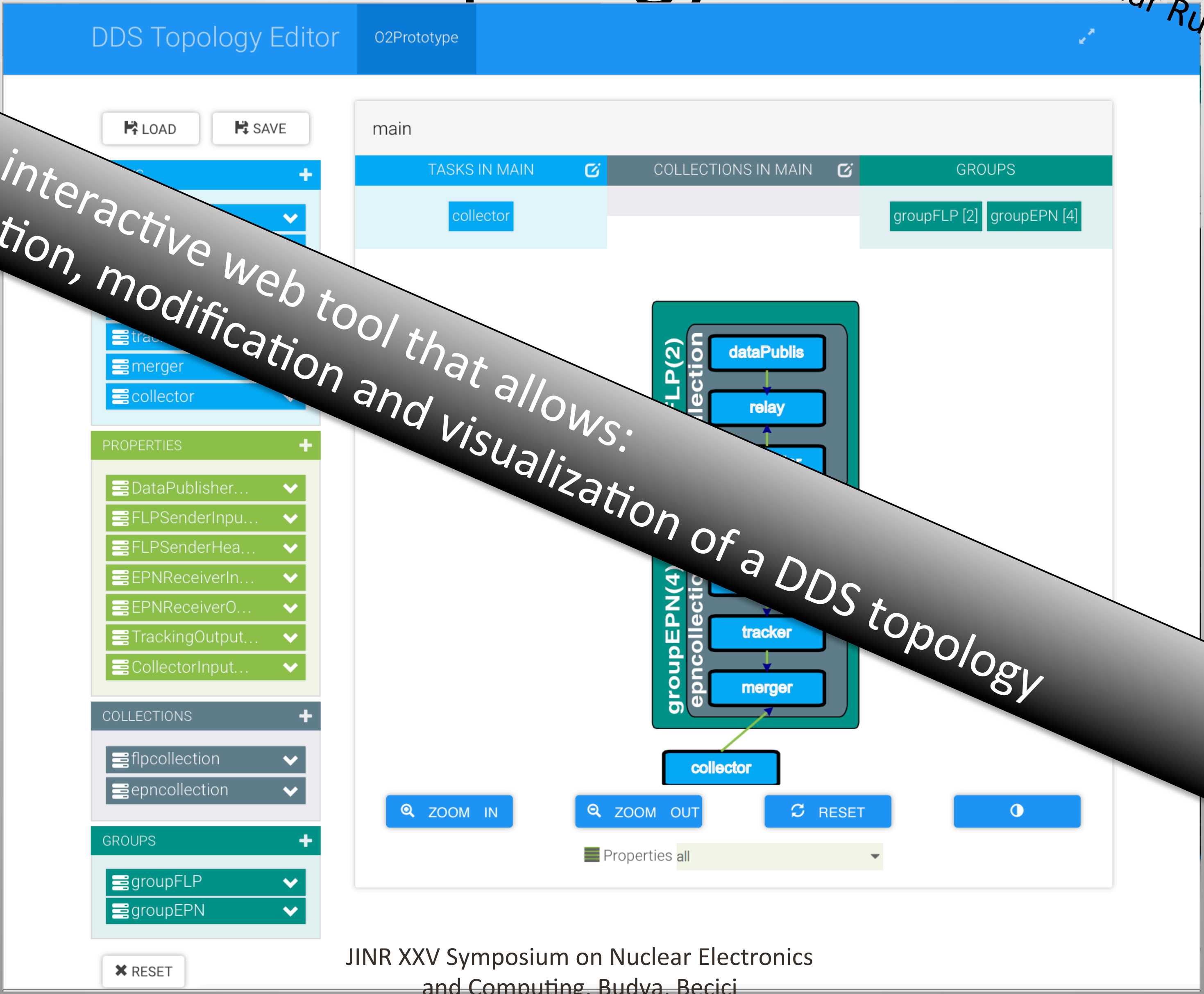
DDS-Topology Editor

Alexey Rybalchenko
Aleksandar Rusinov



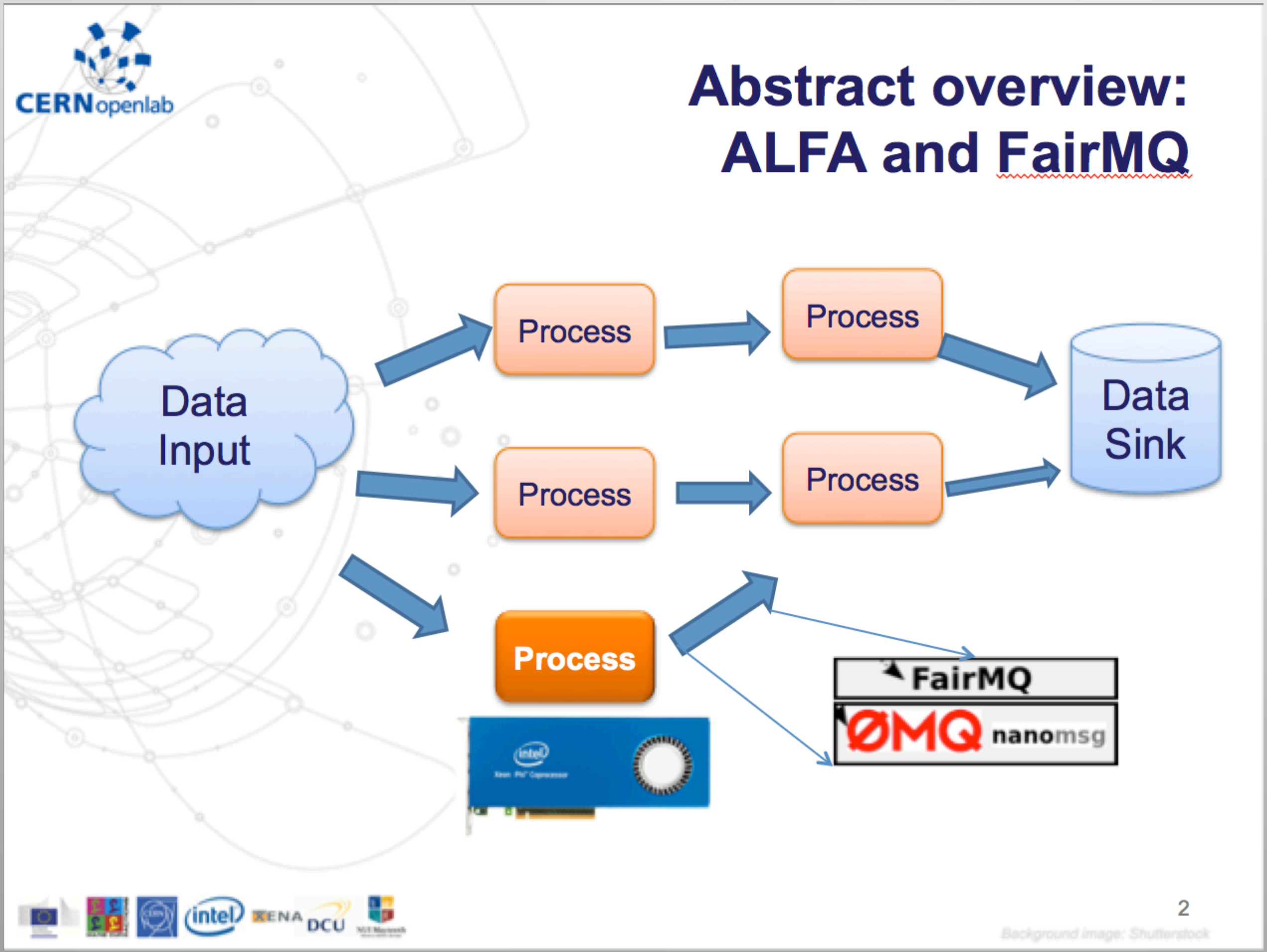
DDS-Topology Editor

Alexey Rybalchenko
Aleksandar Rusinov



An interactive web tool that allows:
creation, modification and visualization of a DDS topology

Xeon Phi



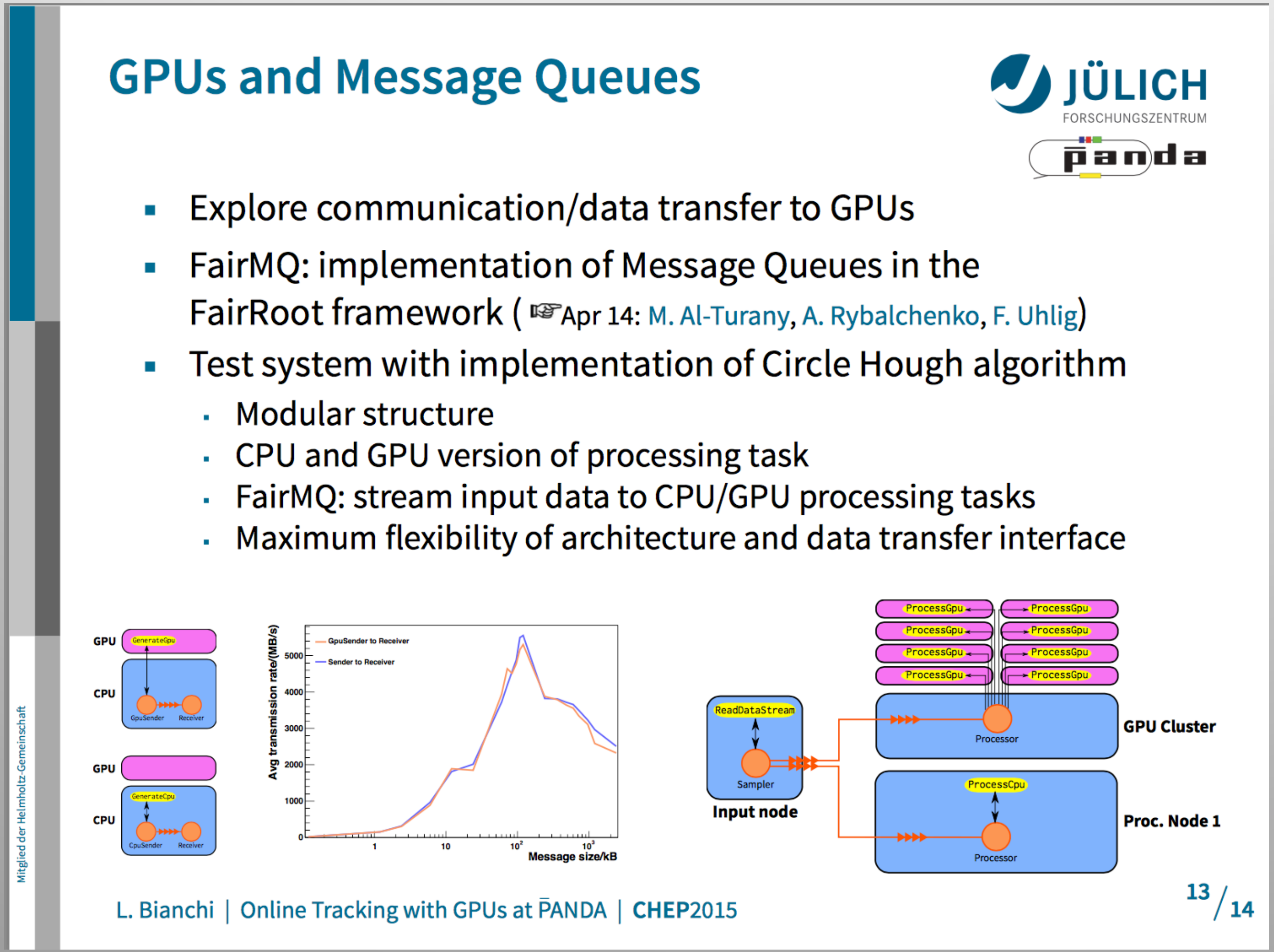
Abstract overview: ALFA and FairMQ

Aram SANTOGIDIS

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

GPUs in ALFA

Ludovico BIANCHI



GPUs and Message Queues

JÜLICH FORSCHUNGSZENTRUM

- 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

Mitglied der Helmholtz-Gemeinschaft

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

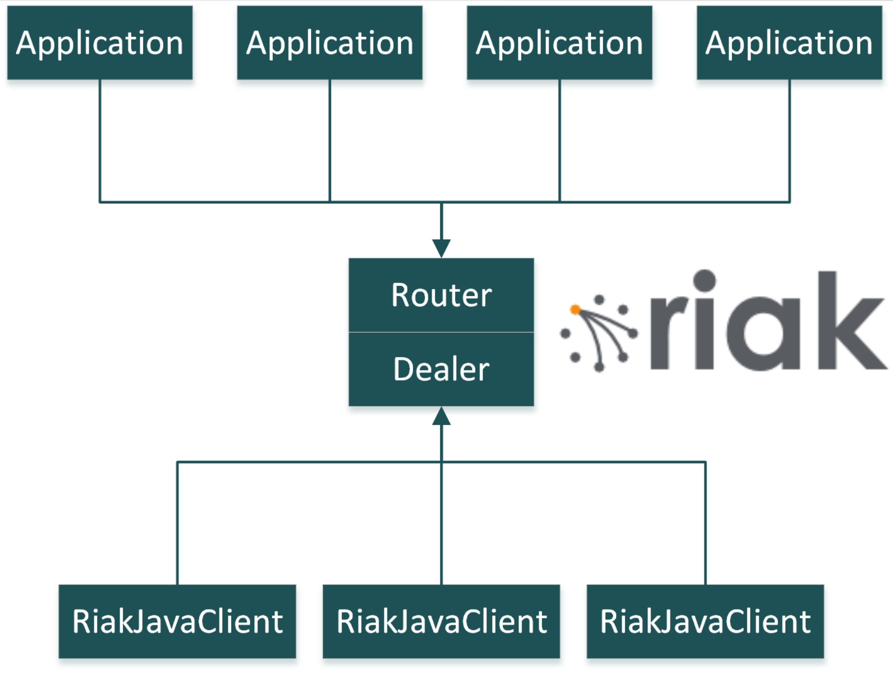
13/14

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

Parameter management

Distributed Model based on Riak

- high availability
- scalability
- fault tolerance
- configurable



Two storage back-ends were tested:

- Bitcask
  - best latency
  - nodes out-of-memory
- LevelDb
  - similar performance
  - compressed storage

Message-Queue based concept make it possible to use directly the native Java client of Riak

ALICE

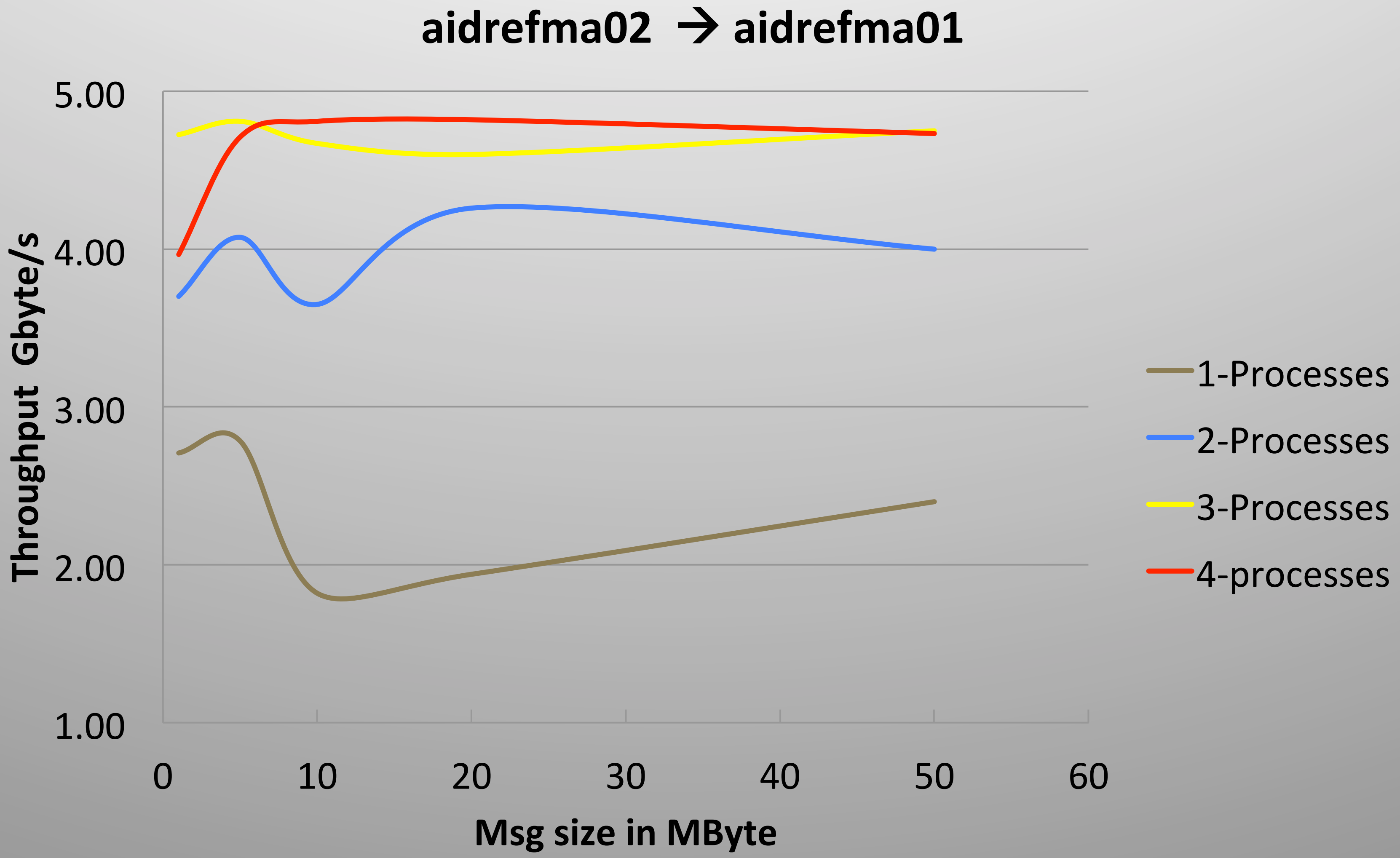
FAIR

Is the data processing strategy feasible?

Can we create a small scale but yet realistic processing topology ?

PROTOTYPE

Running the Zero MQ performance test on the DAQ test cluster (40 GbE)



PROTOTYPE

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

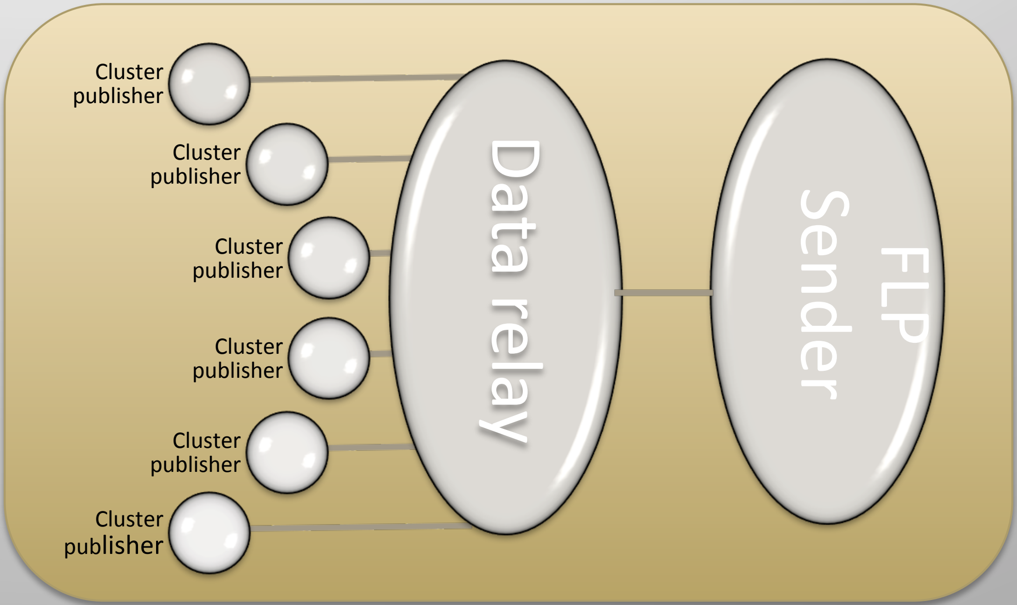
PROTOTYPE

Matthias Richter

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

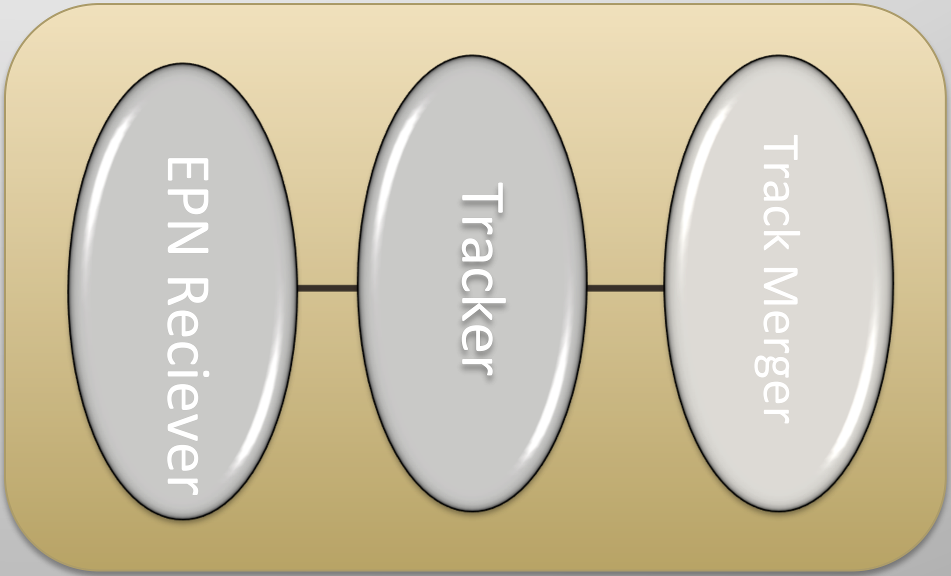
PROTOTYPE

Matthias Richter

EPN devices

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

28+28+28 = 84 processes

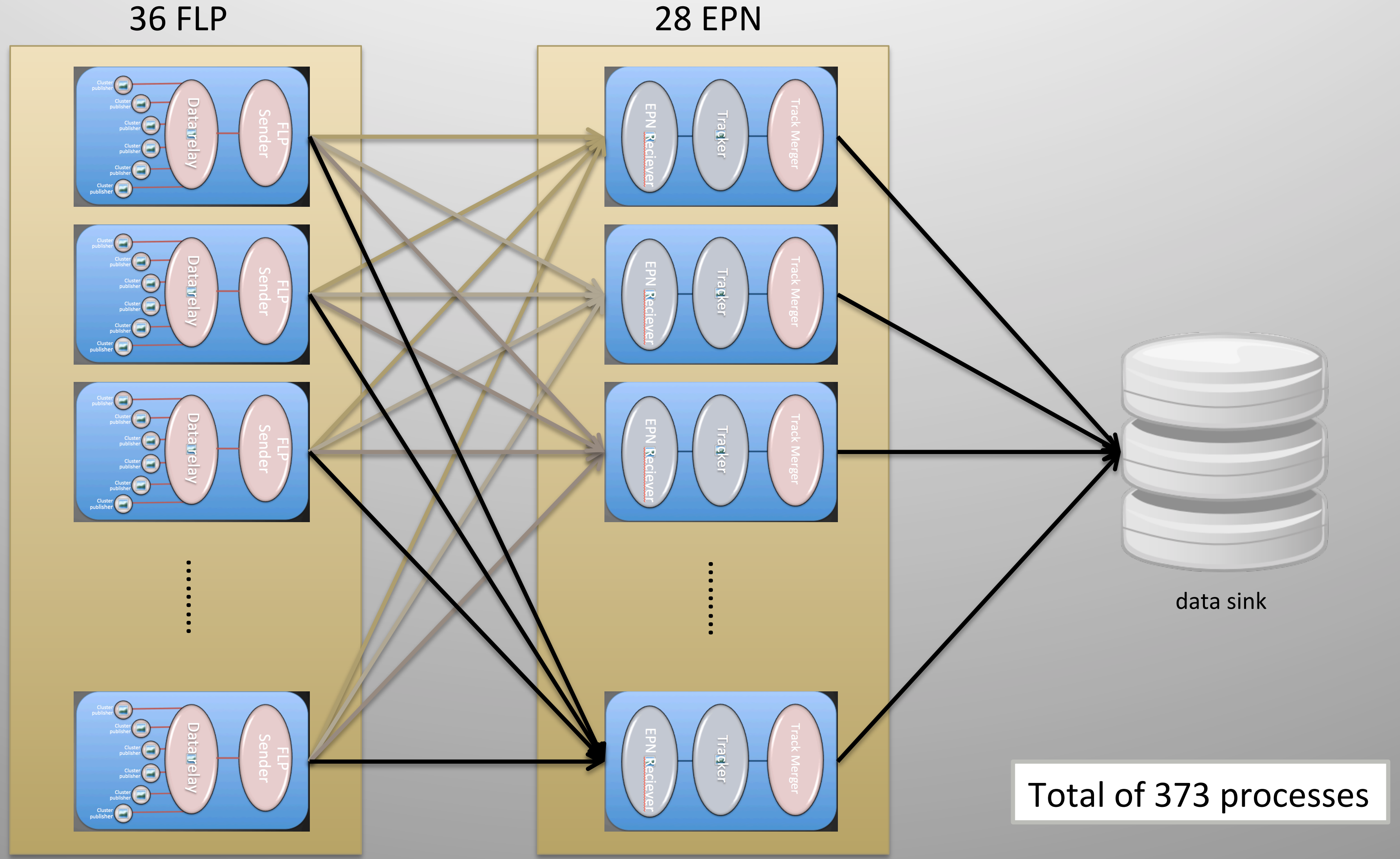


EPN: Event Processing Node

PROTOTYPE

Implementing the TPC reconstruction

Matthias Richter



Total of 373 processes

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

Results

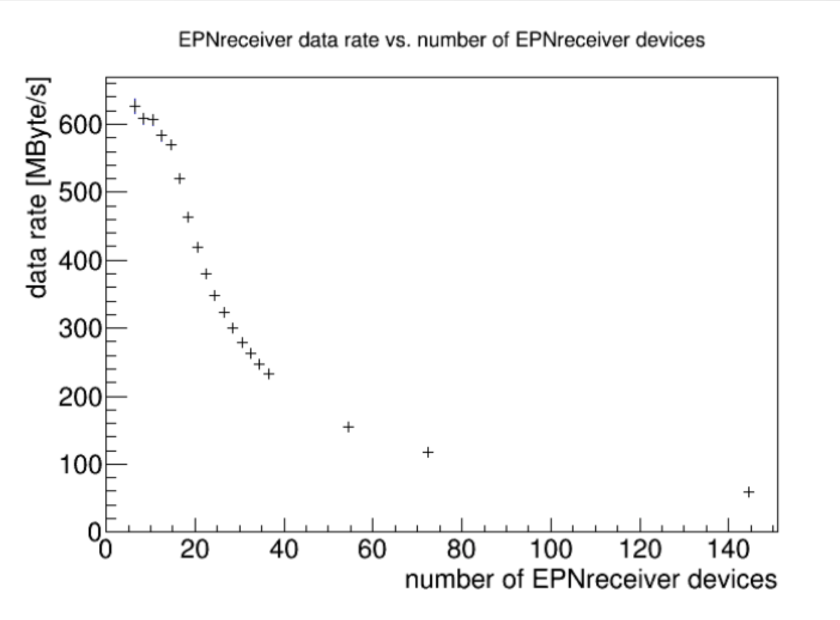
- The topology is processing aggregated size of 1.6 GByte/s (limited by the cluster publishers)
- FLP to EPN data transportation prove to fulfill the requirement
- Efficient process scheduling and deployment system tested with the prototype
- System is ready for larger test

PROTOTYPE

Matthias Richter

Results: EPN

- 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

Release 0.8 (Feb. 15)

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

ØMQ
intel
Xeon
Processor
XILINX
SPARTAN

Another one is under development by the original author of ZeroMQ

- 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.