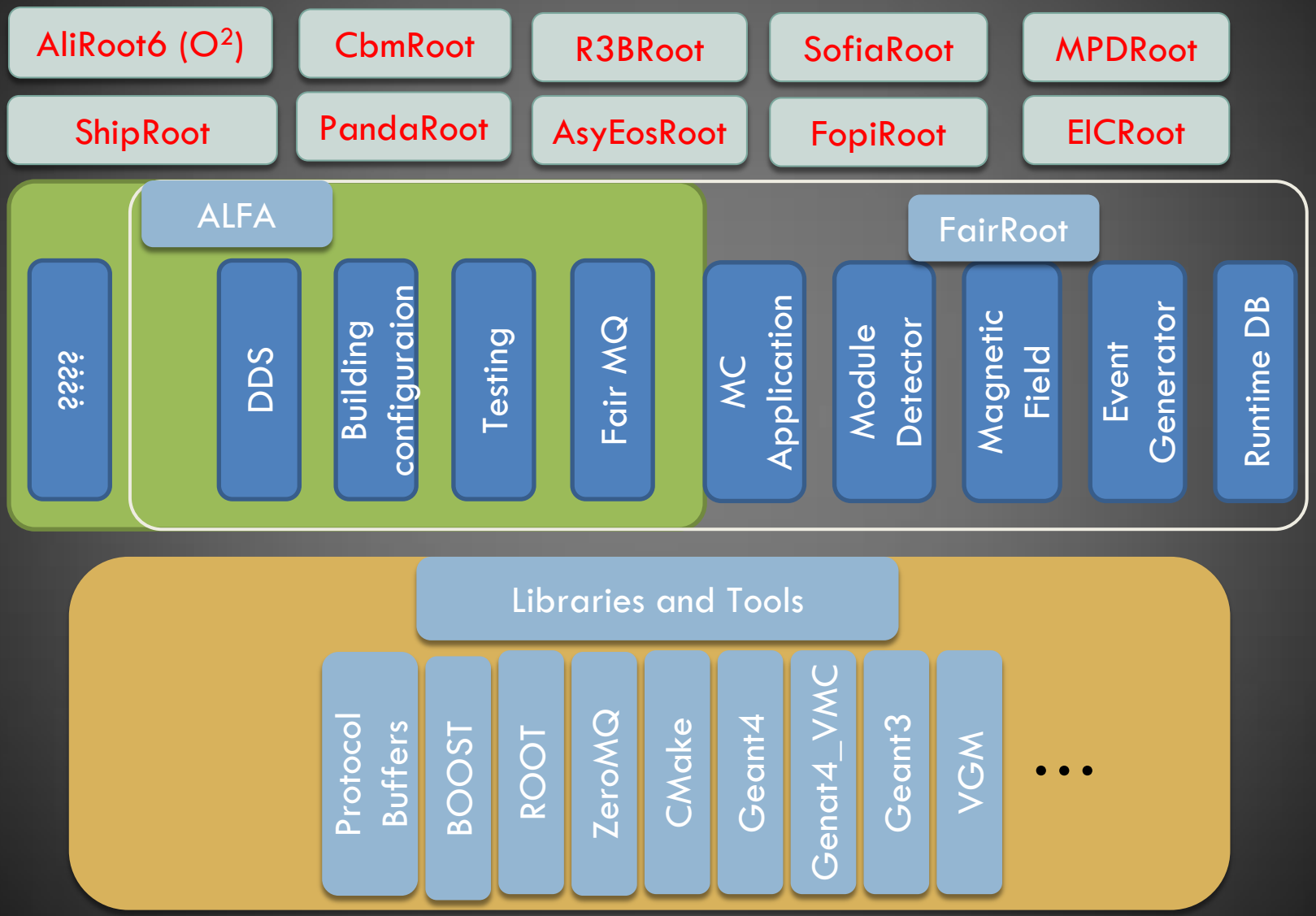
ALFA: The new ALICE-FAIR software framework

Mohammad Al-Turany
GSI-ExpSys/CERN-PH

ALFA

- A tools set library 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



ALFA, FairRoot and co. at CHEP15

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

Ludovico BIANCHI: Online tracking with GPUs at PANDA
https://indico.cern.ch/event/304944/session/1/contribution/363

Dmytro KRESAN : Online/Offline reconstruction of trigger-less readout in the R3B experiment at FAIR
https://indico.cern.ch/event/304944/session/1/contribution/425

AliRoot6 (O^2) | PandaRoot | R3BRoot | SofiaRoot | MPDRoot

ShipRoot | CbmRoot | AsyEosRoot | FopiRoot | EICRoot

Florian UHLIG: New developments in the FairRoot framework https://indico.cern.ch/event/304944/session/2/contribution/258

Tobias STOCKMANNS: Continuous Readout Simulation with FairRoot on the Example of the PANDA Experiment
https://indico.cern.ch/event/304944/session/2/contribution/319

Aram SANTOGIDIS: Optimizing the transport layer of the ALFA framework for the Intel Xeon Phi co-processor
https://indico.cern.ch/event/304944/session/9/contribution/27

Strategy

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

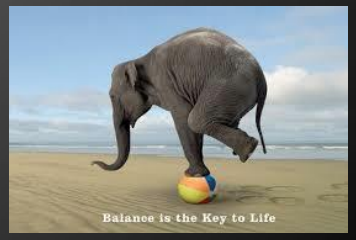
Scalability through multi-processing with message queues?

Each process assumes limited communication and reliance on other processes.

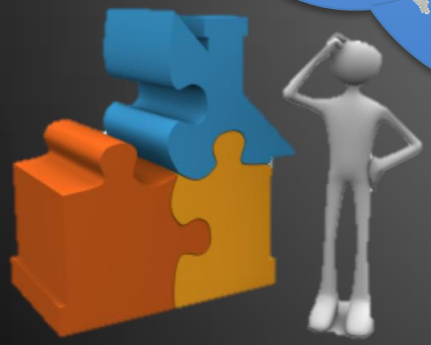
- No locking, each process runs with full speed
- Easier to scale horizontally to meet computing and throughput demands (starting new instances) than applications that exclusively rely on multiple threads which can only scale vertically.

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.



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

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: Basic concepts

- Implements a single-responsibility-principle command line tool-set and APIs,
- Treats users' tasks as black boxes,
- Doesn't depend on RMS (provides deployment via SSH, when no RMS is present),
- Supports workers behind FireWalls,
- Doesn't require pre-installation on WNs (Worker nodes),
- Deploys private facilities on demand with isolated sandboxes,
- Provides a key-value properties propagation service for tasks,
- Provides a rules based execution of tasks.
- ...

http://dds.gsi.de

Tests of ALFA devices via DDS

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	~ $77x10^6$

- FLP(First Level Processer), EPN (Event Processor Event)
- N –To-N connections
  - Each EPN need to connect to each FLPs (Time frame building)
  - Each FLP need to connect to all EPNs (Heartbeat signal)
- Throughout tests only one DDS commander server was used. In the future we will support multiple distributed servers.
- Our current short term target is to manage 100K devices and keep startup time of the whole deployment within 10-50 sec.

* 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.

Backup

DDS status

- Last stable release - DDS v0.8 (2015-02-17, http://dds.gsi.de/download.html),
- Home site: http://dds.gsi.de
- User's Manual: http://dds.gsi.de/documentation.html
- Continues integration: http://demac012.gsi.de:22001/waterfall
- Source Code:
  https://github.com/FairRootGroup/DDS
  https://github.com/FairRootGroup/DDS-user-manual
  https://github.com/FairRootGroup/DDS-web-site
  https://github.com/FairRootGroup/DDS-topology-editor

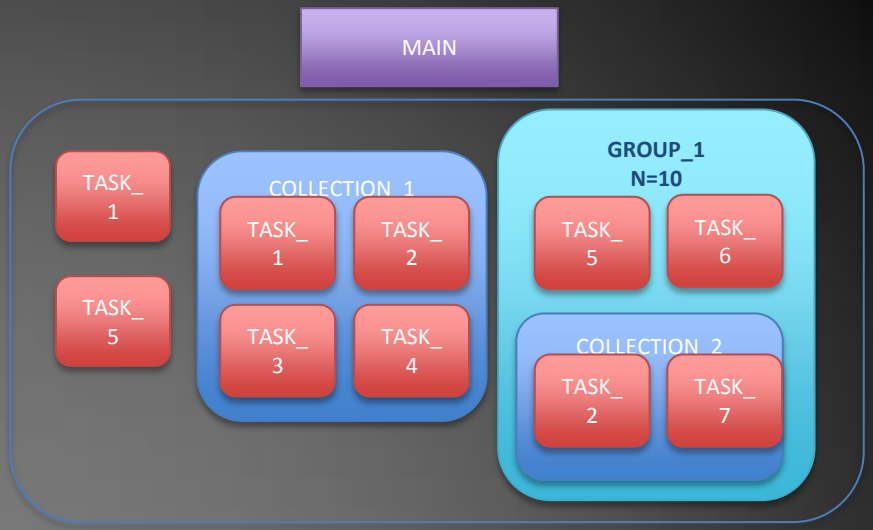
Related talks

- Florian UHLIG: New developments in the FairRoot framework
  https://indico.cern.ch/event/304944/session/2/contribution/258
- 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
- Tobias STOCKMANNS: Continuous Readout Simulation with FairRoot on the Example of the PANDA Experiment
  https://indico.cern.ch/event/304944/session/2/contribution/319

Related talks

- Ludovico BIANCHI: Online tracking with GPUs at PANDA
  https://indico.cern.ch/event/304944/session/1/contribution/363
- Dmytro KRESAN : Online/Offline reconstruction of trigger-less readout in the R3B experiment at FAIR
  https://indico.cern.ch/event/304944/session/1/contribution/425
- Aram SANTOGIDIS: Optimizing the transport layer of the ALFA framework for the Intel Xeon Phi co-processor
  https://indico.cern.ch/event/304944/session/9/contribution/27

Elements of the topology



Task

* A task is a single entity of the system.
* A task can be an executable or a script.
* A task is defined by a user with a set of props and rules.
* Each task will have a dedicated DDS watchdog process.

Collection

* A set of tasks that have to be executed on the same physical computing node.

Group

* A container for tasks and collections.
* Only main group can contain other groups.
* Only group define multiplication factor for all its daughter elements.