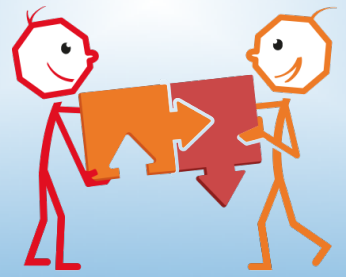
ALFA: ALICE-FAIR software framework

Mohammad Al-Turany

GSI scientific computing

A new system developed in common by FairRoot Group (GSI), FAIR experiments and ALICE O2



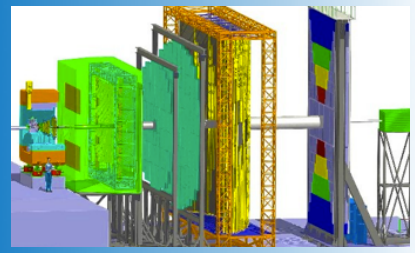
Why?

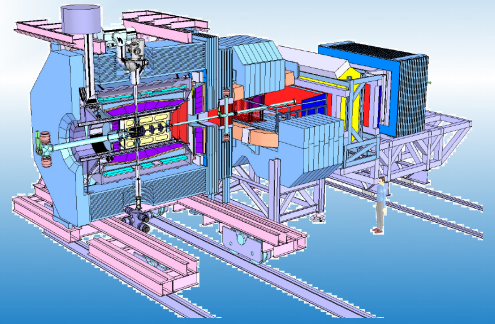
Same requirements

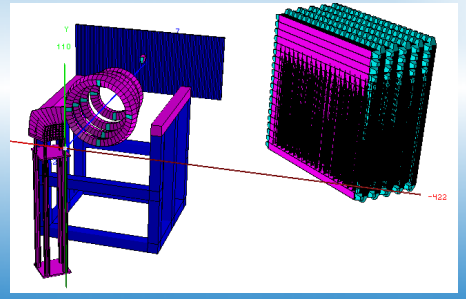
Massive data volume reduction (1-3 TByte/s input)

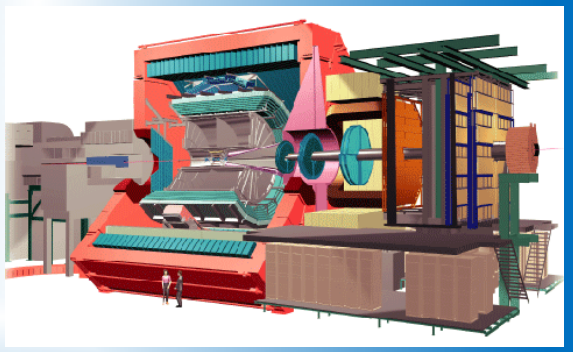
Data reduction by (partial) online reconstruction

Online reconstruction and event selection





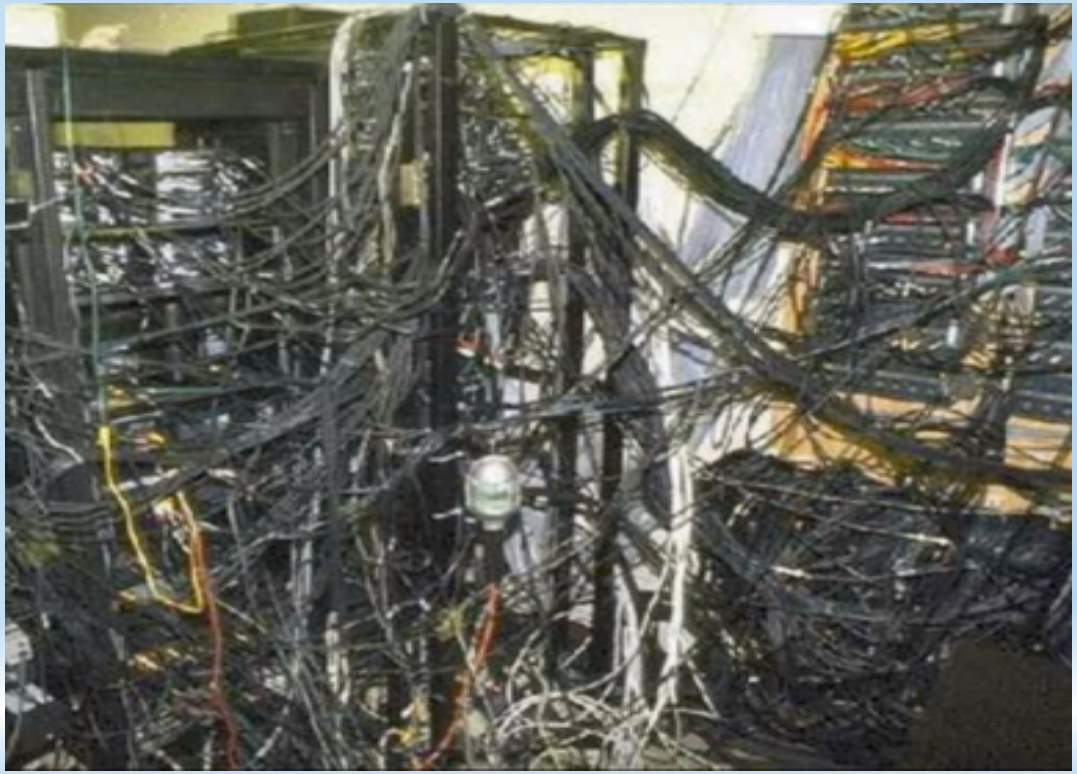




What is new in ALFA compared to AliRoot, FairRoot, …etc

Has data-flow based model: Message Queues based multi-processing

Works across all types of networks

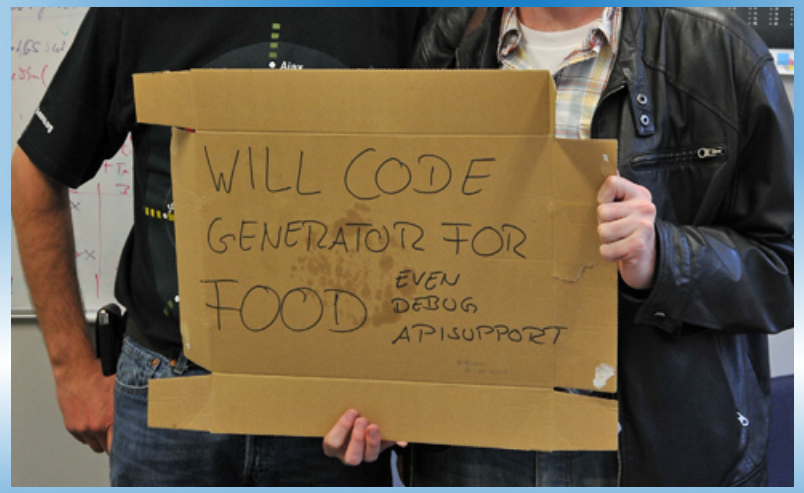


Scale linearly to any number of cores

ALICE

FAIR

Can be easily used, adapted and extended by typical programmers/physicist .



Actor Model

Standalone processes ("devices") perform a task (e.g. track finding) and communicate with each other via messages (mediated by a queue).

Actor Model

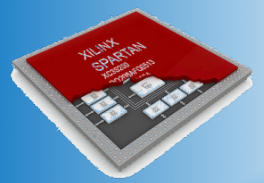
- No locking, each process runs with full speed
- Easier to scale horizontally to meet computing and throughput demands (start/add new instances)

Right tools for the right job!

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)

Message format ?



The framework does not impose any format on messages.

It supports different serialization standards

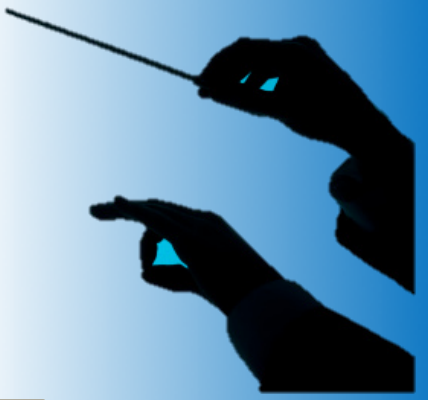
- BOOST C++ serialization
- Google's protocol buffers
- ROOT
- Flatbuffers
- MessagePack
- User defined

Extendable, plugin based system to Configure and Control devices.

Different topologies of tasks can be adapted to the problem itself and the hardware capabilities

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 (graph) files

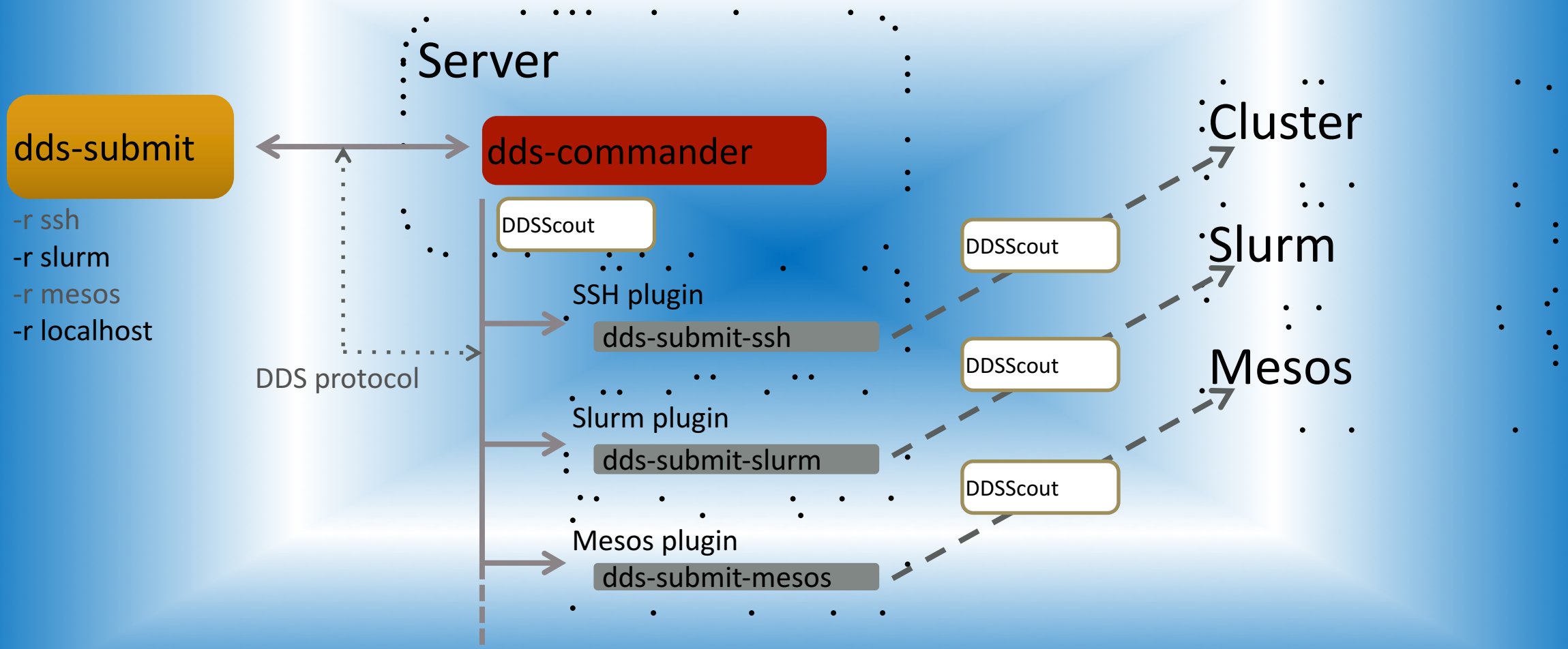
Users are provided with a WEB GUI to create topology (Can be created manually as well).

http://dds.gsi.de/

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 with only outgoing connection,
- doesn't require pre-installation on WNs,
- deploys private facilities on demand with isolated sandboxes,
- provides a key-value properties propagation service for tasks,
- provides a rules based execution of tasks.

DDS: RMS plug-in architecture



ALFA Transport layer: FairMQ

- Hide all the sockets and message transport operation from the user
- Allow non-expert to write messaged based code without going into the details of the transport or the system below
- Offer a **clean** and **maintainable** and **extendable** interface to the existing different data transport (ZMQ, nanomsg, shared Memory,..etc)

ALFA Transport layer: FairMQ

- Allow usage of combinations of transport layers in one device in a transparent way

- Any device/channel should be able to switch transport only via configuration, without modifying device/user code -> same API.

User interface

- User code can be implemented as devices (sub-classes of FairMQDevice), moreover a main function has also to be implemented (executable)
- User can also use the simplified interface with common main and callback API

How to switch form root single-core processing (FairRoot/Root Tasks) to FairMQ multi-core pipeline processing

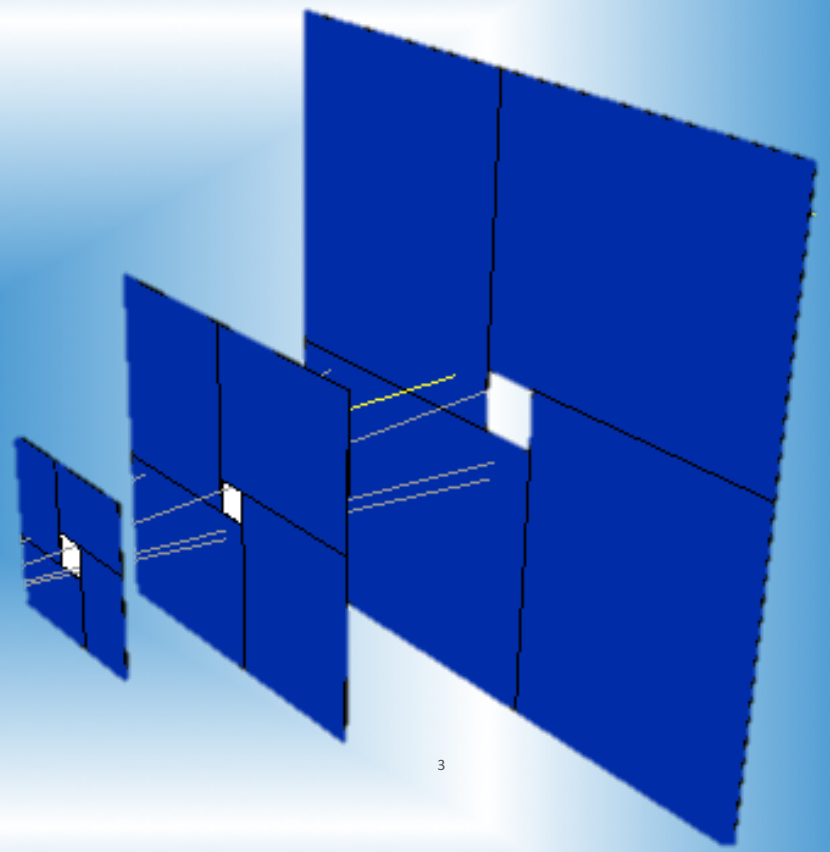
FairRoot/Examples/MQ/9-PixelDetector

- Detector simulation,
- Digitization,
- reconstruction (hit finding, tracking, track fitting),
- Shows how to switch from root single-core processing to FairMQ multi-core pipeline processing.

https://github.com/FairRootGroup/FairRoot/tree/master/examples/MQ/9-PixelDetector

fairroot/examples/MQ/9-PixelDetector

- 3 stations with 4 rectangular sensor each:
  - size: 5x 5cm^2, inner hole: 1x1cm^2, at z = 5cm;
  - size: 10x10cm^2, inner hole: 1x1cm^2, at z = 10cm;
  - size: 20x20cm^2, inner hole: 2x2cm^2, at z = 20cm;
- each sensor divided into pixels (0.01x0.01cm^2), that are grouped into FE modules (110 pixels x 116 pixels)

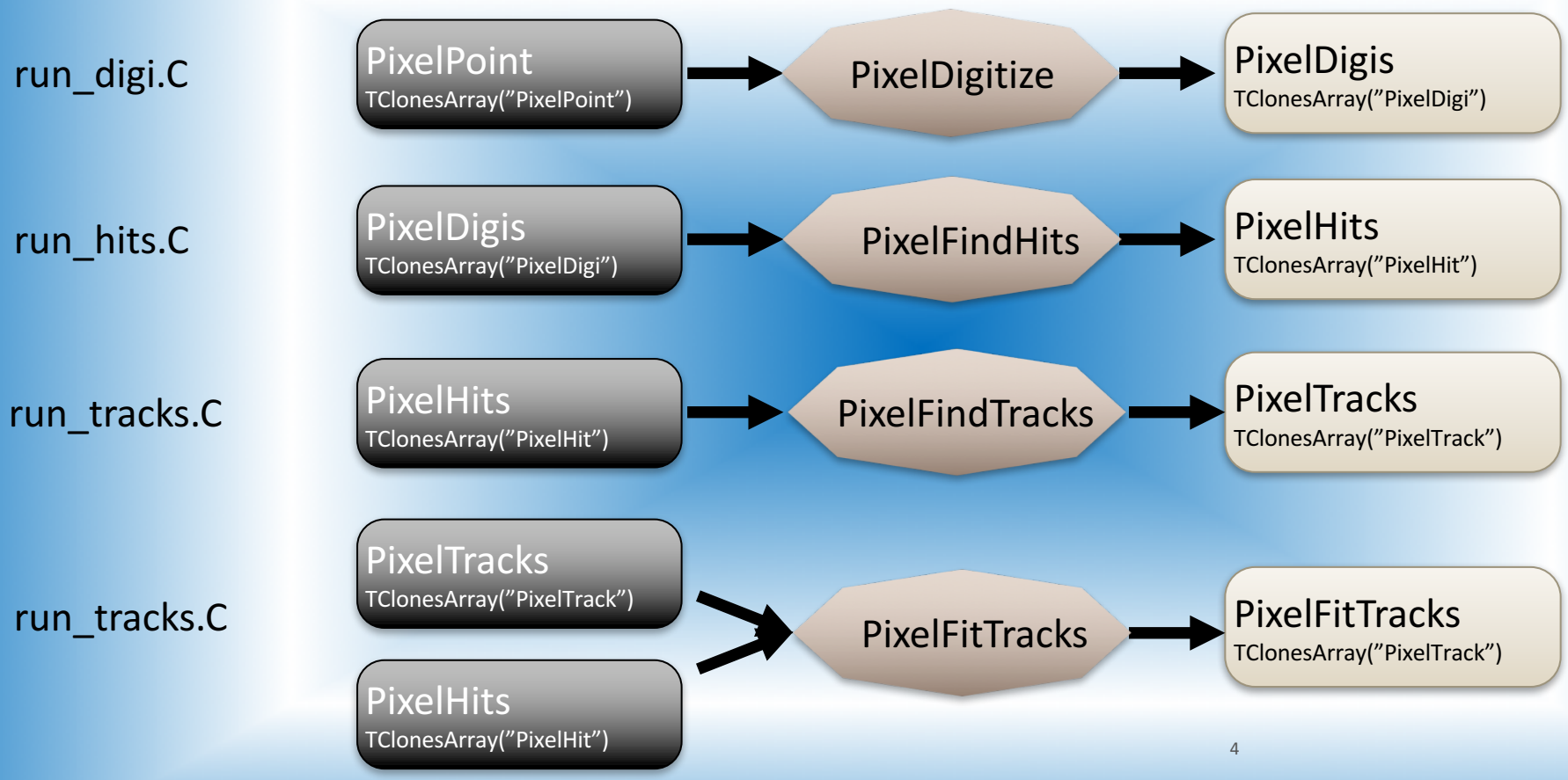


FEs numbering on one sensor

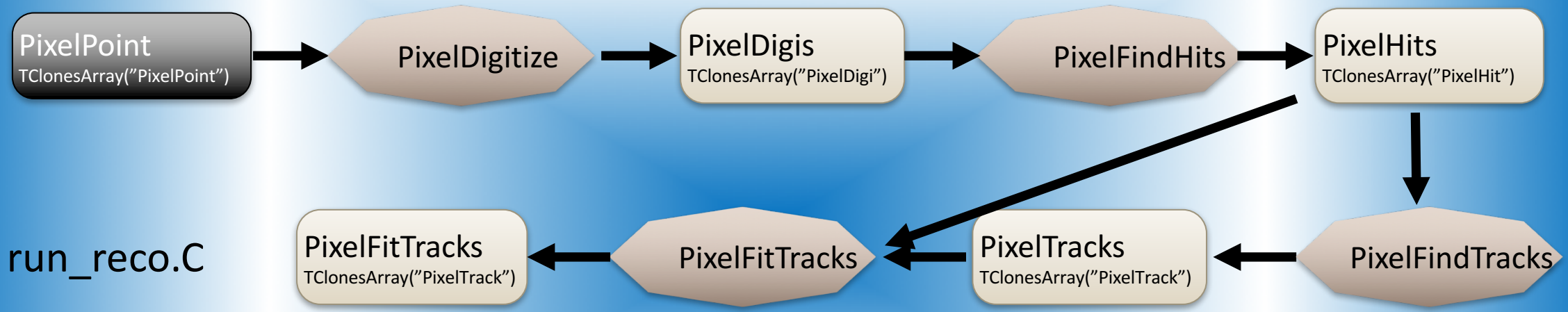
…					
FE 5	…				
FE 4	FE 68	…			
FE 3	FE 67	FE 131	…		
FE 2	FE 66	FE 130	FE 194	…	
FE 1	FE 65	FE 129	FE 193	FE 257	…

https://github.com/FairRootGroup/FairRoot/tree/dev/examples/MQ/9-PixelDetector

data classes, tasks and macros

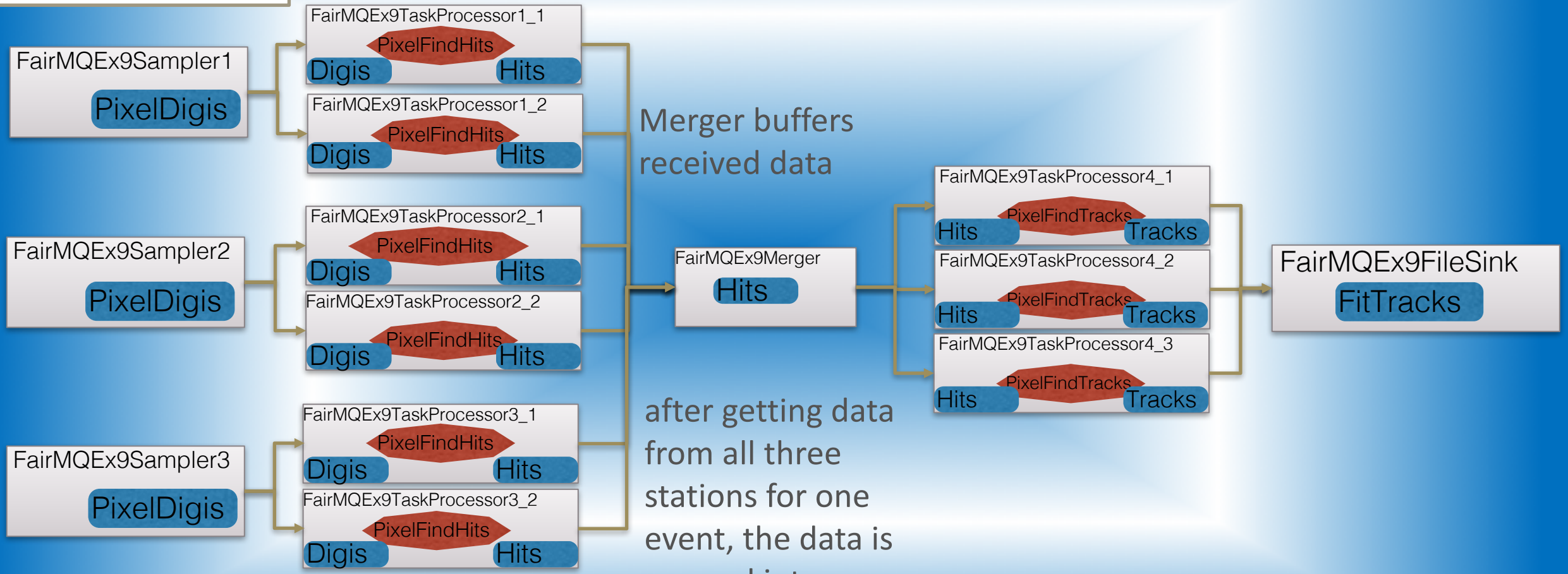


data classes, tasks and macros



Example topology

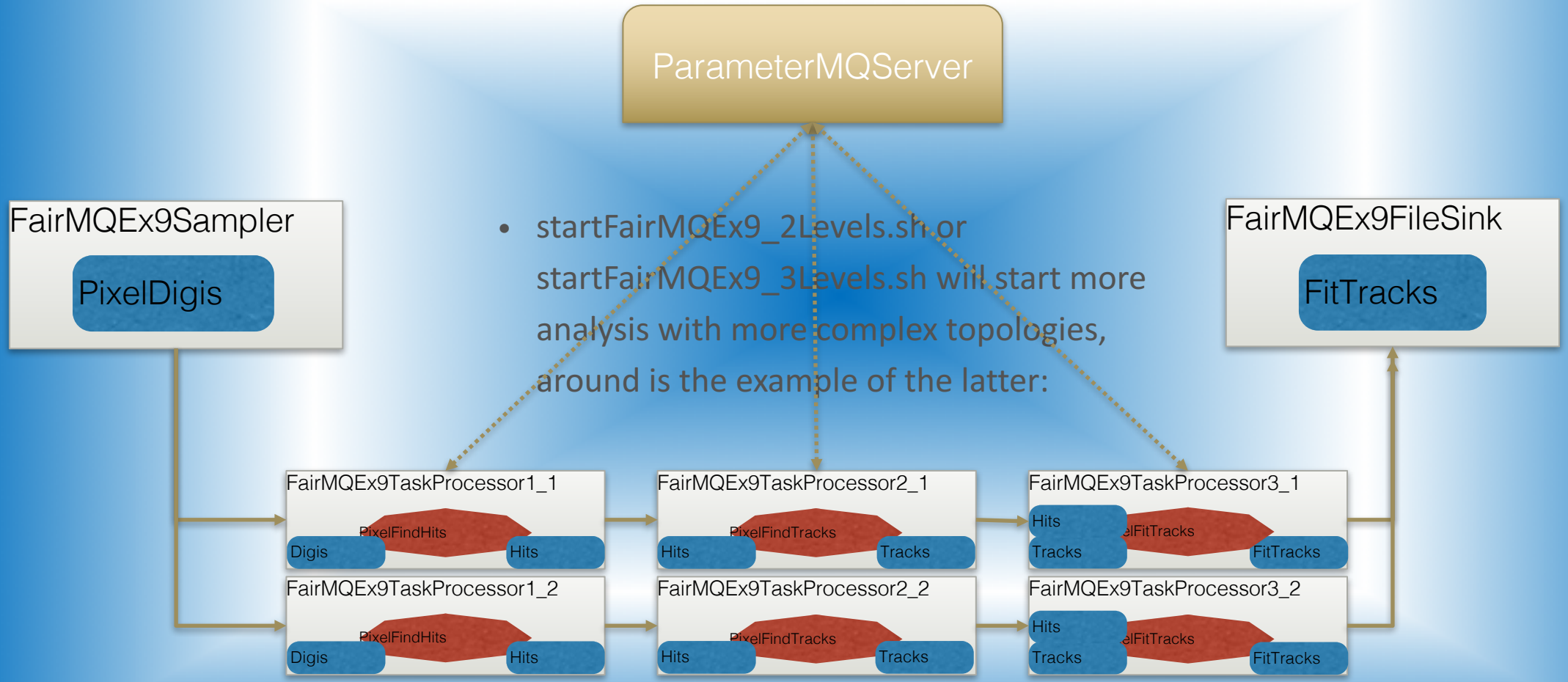
Each sampler reads from file with data from one station



Merger buffers received data

after getting data from all three stations for one event, the data is merged into one event and sent

Other topologies:



- startFairMQEx9_2Levels.sh or startFairMQEx9_3Levels.sh will start more analysis with more complex topologies, around is the example of the latter:


AH
I WONDER WHATS
NEXT

Vectored IO

- Vectored IO (scatter-gather) is under investigation
  - A single call reads data from multiple buffers and writes it to a single data stream, or reads data from a data stream and writes it to multiple buffers.
- It will be on the FairMQ level
  - Could re-use the FairMQParts interface but no-multipart in the sense of ZeroMQ

RDMA

- Work on RDMA transport is ongoing
- Different options are under investigation

Summary

- ALFA provides an environment to write message passing processes, abstracting away many of the implementation details
- Provides utilities to deploy topologies (via DDS) to many resource managements systems
- Interface to access condition data (Parameter Manager)
- Plugin based system to configuration, control and monitoring of user processes