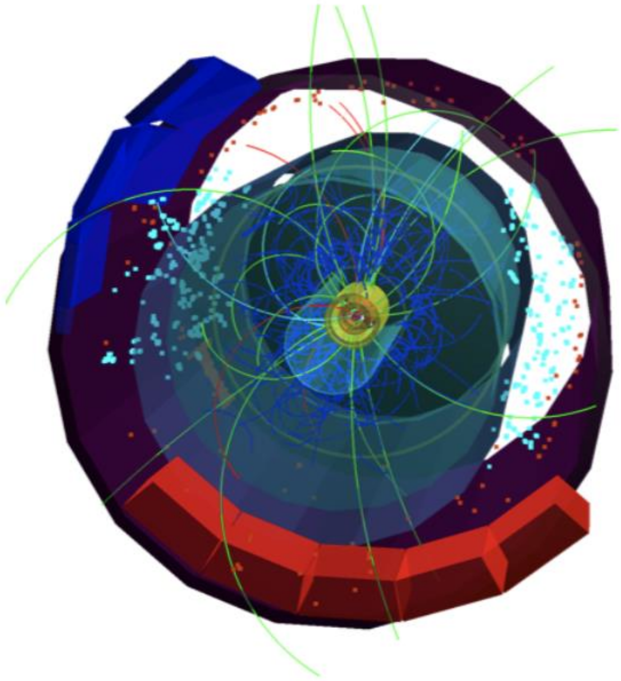
# Analysis in LHC RUN3 (Alice case)

M. Al-Turany, GSI/IT

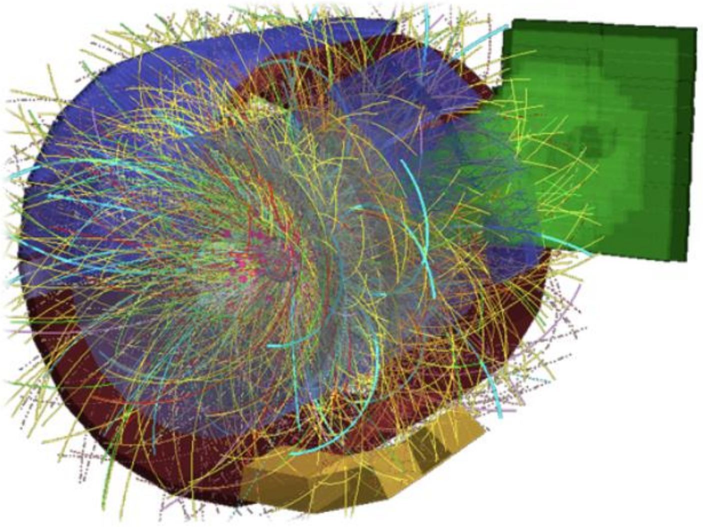
Thanks to:

Giulio Eulisse, Peter Hristov, Ruben Shahoyan, Thorsten Kollegger, Killian Schwarz

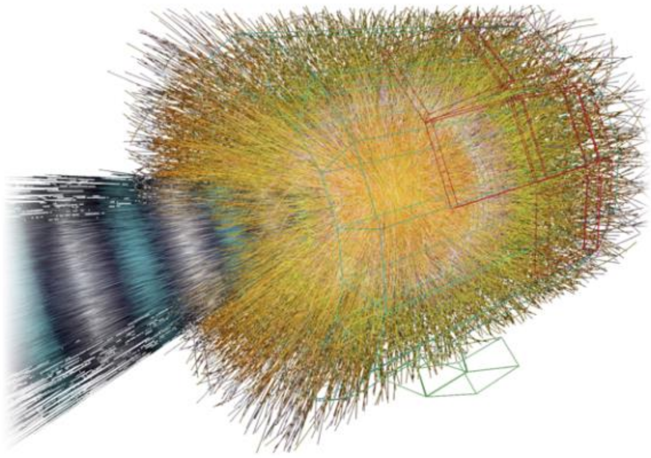
# Alice in RUN2
# about O(1) kHz single events
# more than 4 Gbytes/s to Storage



*p - p*



*p - Pb*



*Pb - Pb*

# Analysis in RUN2:

- Organized analysis
- Event-oriented data model: trees of ESD & AOD/delta AOD, but also kinematics, ESD friends, track references, tags
  - Access to the different data via handlers
- Possibility to run in local, Proof, GRID, event mixing modes
  - Services: I/O, event loop, merging of results, bookkeeping
  - LEGO trains
- All user code on GitHub (alisw/AliPhysics) and built centrally on CVMFS

# Analysis Trains:

Analysis tasks organized in trains (dependencies, I/O):

- Read data once,
- process many times,
- benefit from common processing

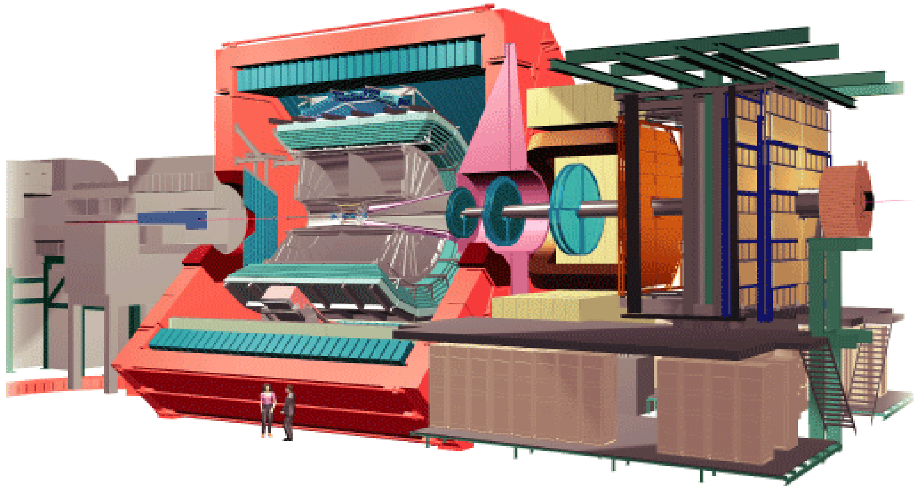
# ALICE Analysis Facilities (Run1/Run2)

- PROOF-based facilities: CERN, Lyon, GSI, Torino and many other places
    - Local data sets
    - Running on native resources
    - Using shared file system
    - Remote access from laptop/desktop

# ALICE Upgrade

- The Inner Tracking System (ITS) will be replaced with a new, high-resolution, low-material detector
- The Time Projection Chamber (TPC) will be upgraded with replacement of the chambers by Gas Electron Multipliers (GEMs) and a new pipelined readout electronics based on a continuous read-out scheme
- The forward trigger detectors and the electronics of the Transition Radiation Detector (TRD), the Time Of Flight (TOF), and several other detectors will be upgraded

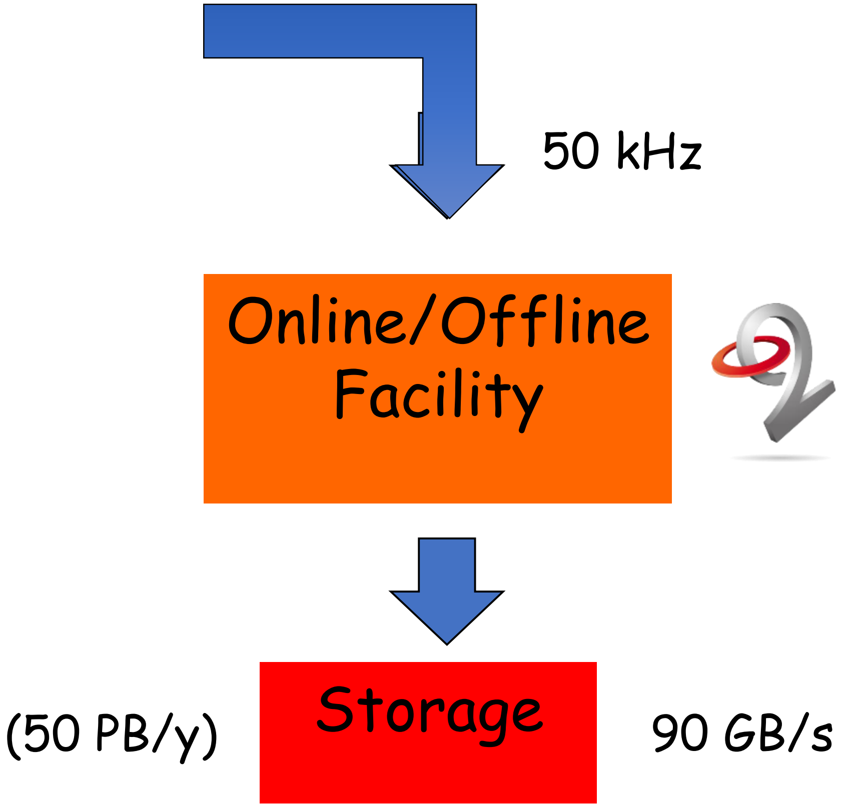
# ALICE Upgrade



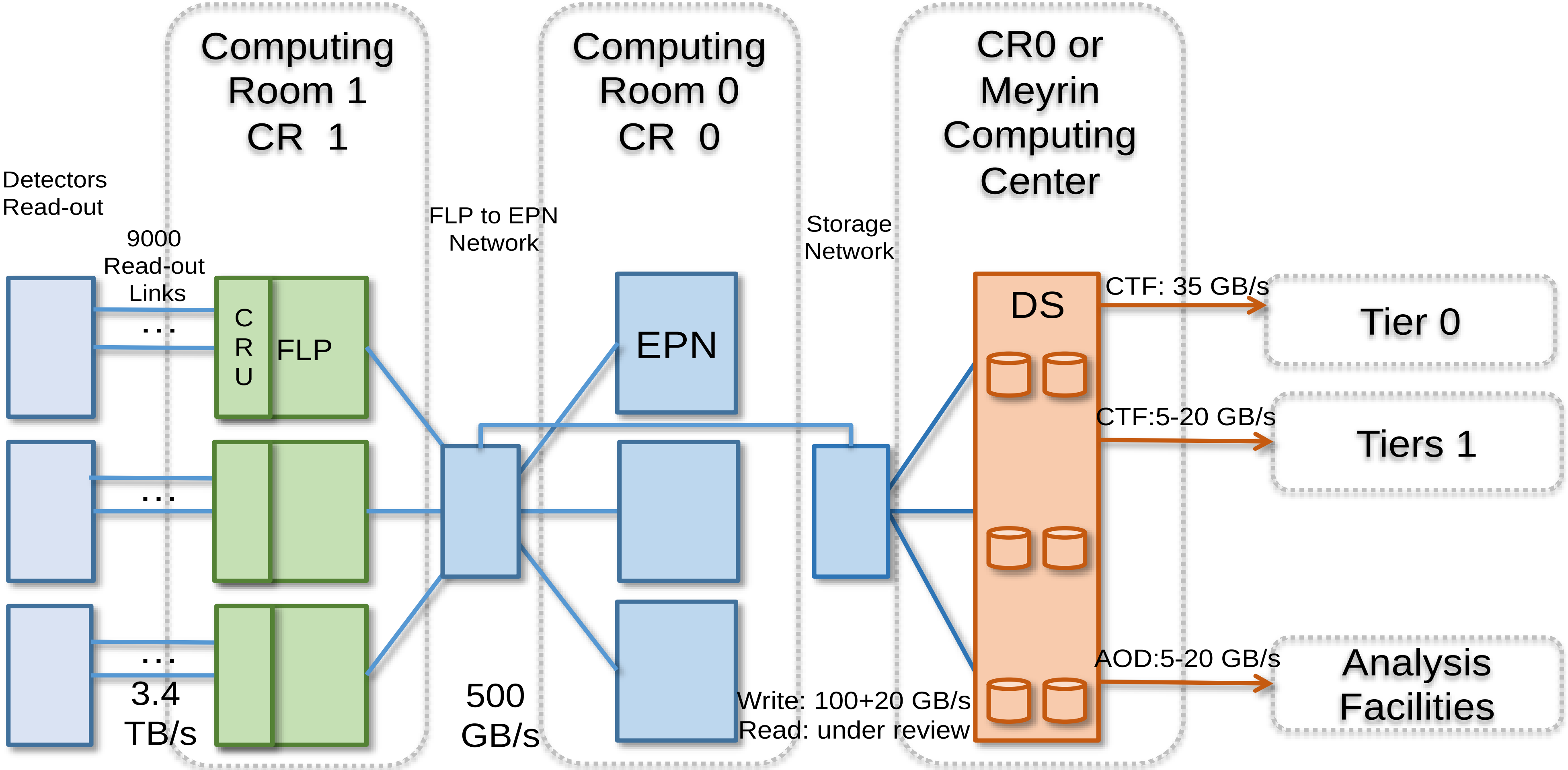
- Aim is to reduce data volume by doing (quasi) online reconstruction
  - Each and every event needs to be processed, no rejection
- High Throughput (and not Performance) Computing problem

- continuous readout
- x50 event rate

3.4 TB/s



# Online Reconstruction:  O2 Facility

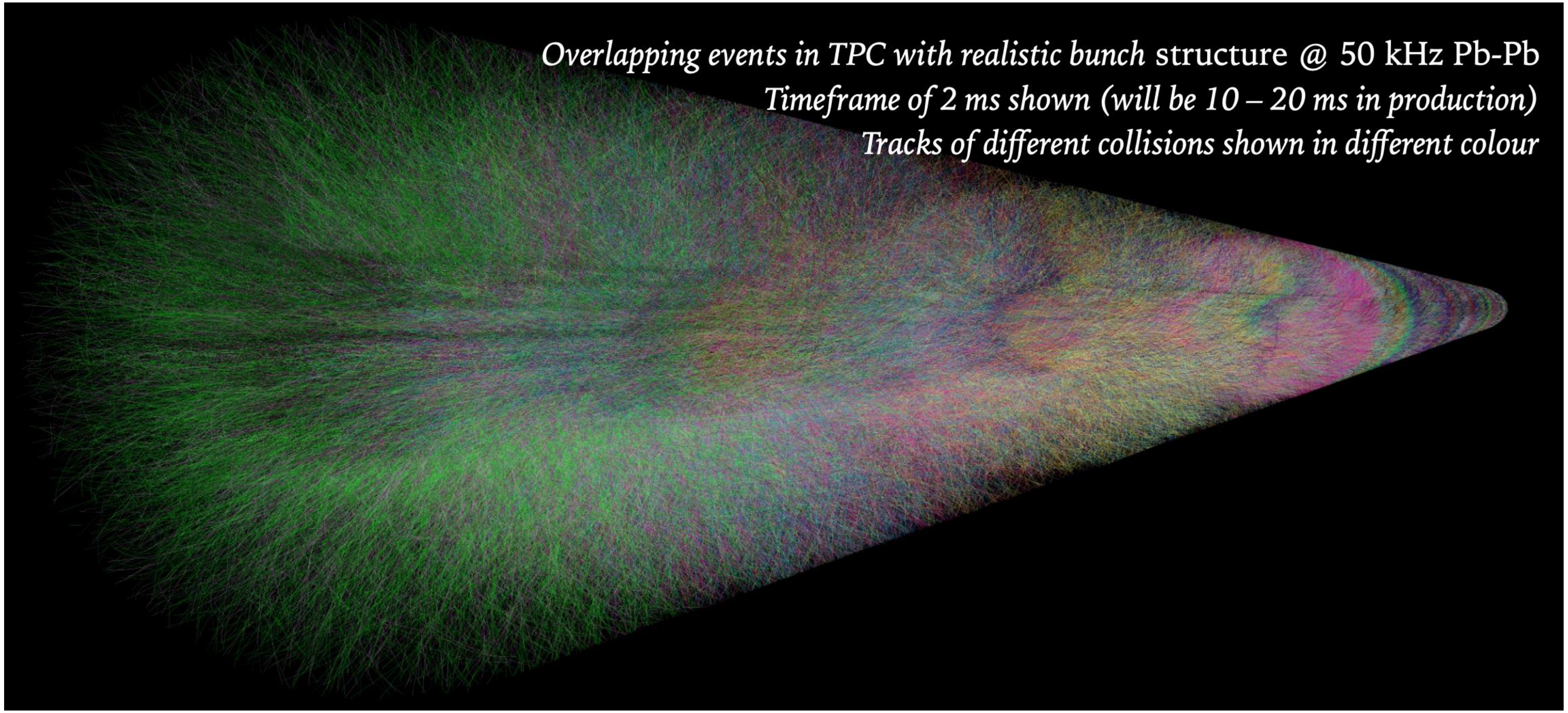


Baseline correction, zero suppression cluster finder.

Data volume reduction by online tracking, Asynchronous processing

Data Storage

# Alice in RUN3
# 50 kHz of continuous readout data.
# 90 Gbytes/s to Storage (50 PB/y)

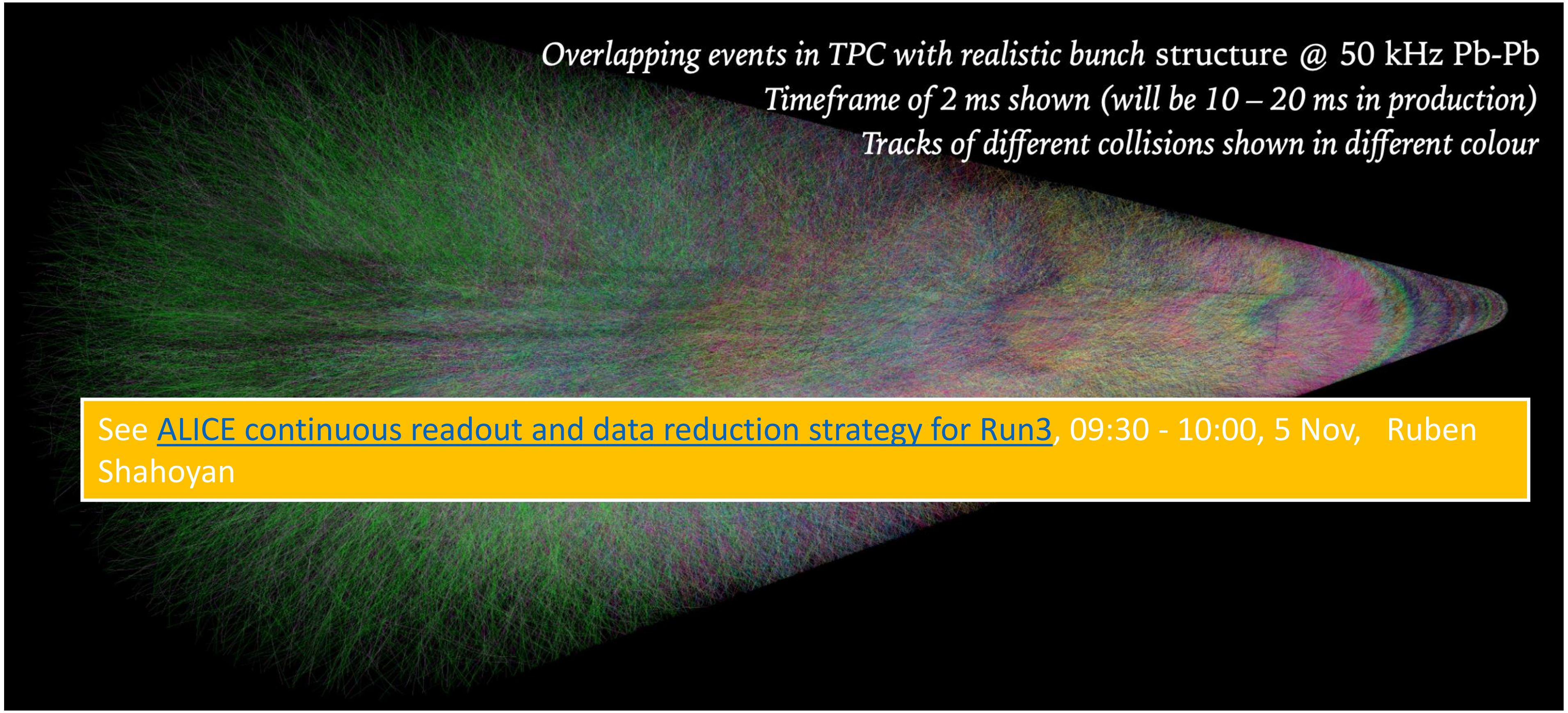


*Overlapping events in TPC with realistic bunch* structure @ 50 kHz Pb-Pb

*Timeframe of 2 ms shown (will be 10 – 20 ms in production)*

*Tracks of different collisions shown in different colour*

# Alice in RUN3
# 50 kHz of continuous readout data.
# 90 Gbytes/s to Storage (50 PB/y)



*Overlapping events in TPC with realistic bunch structure @ 50 kHz Pb-Pb*
*Timeframe of 2 ms shown (will be 10 – 20 ms in production)*
*Tracks of different collisions shown in different colour*

See ALICE continuous readout and data reduction strategy for Run3, 09:30 - 10:00, 5 Nov, Ruben Shahoyan

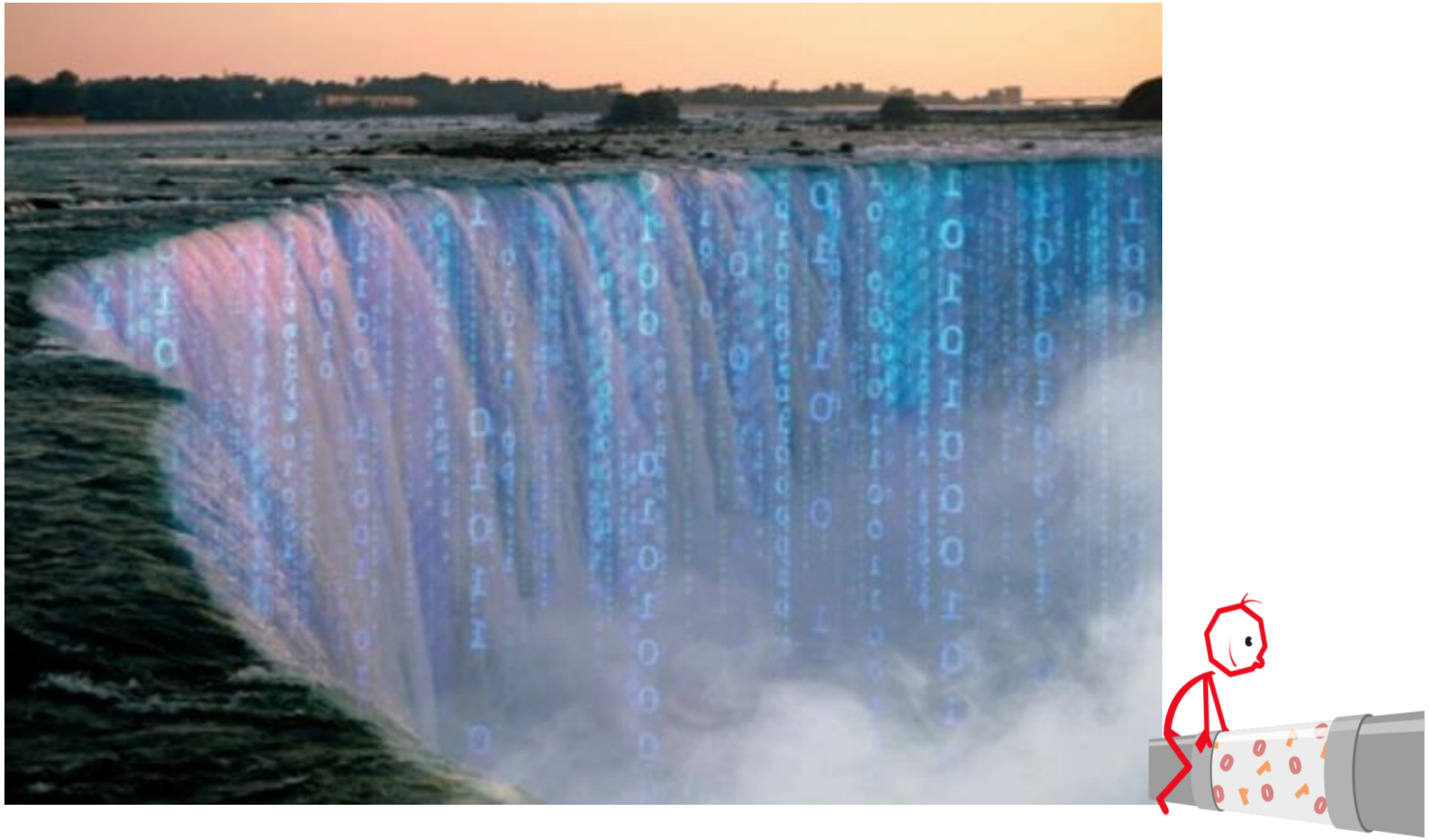
# Compared to RUN2

- ***Reconstruct 50x more*** *events online*
- ***Store 50x*** *more events*
- ***continuous readout (****TPC data* ***)*** *in combination with data coming from triggered detectors.*

# What to do?

ALICE can cope with the challenges of Run3 only by a radical redesign of its software and computing architecture.

# A data-flow based model:



Message Queues based multi-processing

# A data-flow based model:

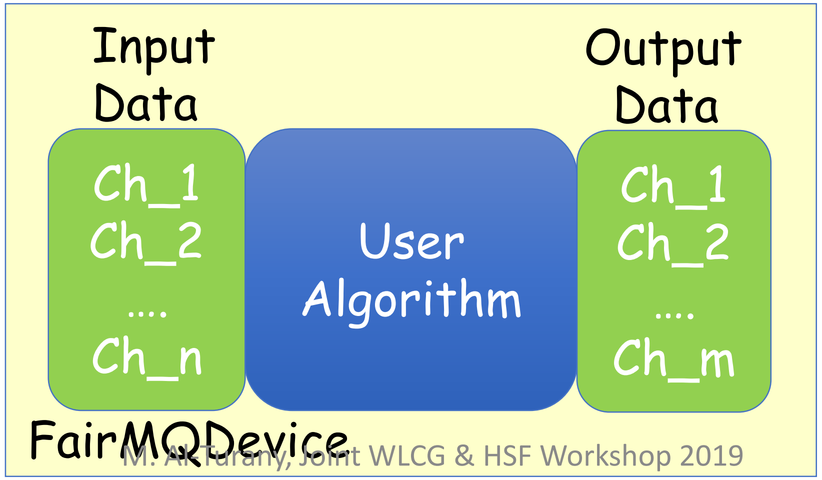
ALFA: A framework for building distributed applications

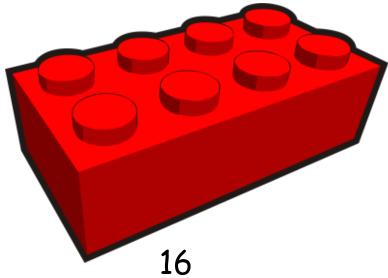
Track 5 - Software Development 11:30 - 11:45

Message Queues based multi-processing

# ALFA building block (FairMQ Devices)

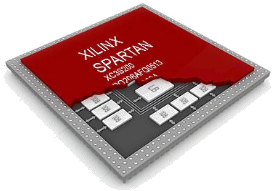
- Message Queues for input/output
- Device takes/passes ownership of data
- Framework user sees only the callback to his algorithm
- Different channels can use different transport engines





# Message format ?

FairMQ does not impose any format on messages.



It supports different serialization standards

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

# Software framework:
# Transport Layer

- Uses FairMQ message passing toolkit (GSI development)
- Abstracts the network fabric
- Defines the core building blocks in terms of devices
- Implements the communication between them

ALFA: A framework for building distributed applications
**Track 5 – Software Development** 11:30 - 11:45

# Software framework:
# O2 Data Model

- ALICE-specific description of the messages between devices
- Computer language agnostic, extensible, efficient mapping of the data objects in shared memory or to the GPU memory
- Supports multiple data formats and serialization methods

# Software framework:
# Data Processing Layer

Data Analysis using ALICE Run3 Framework,
5 Nov, G.Eulisse, T6, 11:45

- Simplifies the life of the end user
- Allows to describe computation as a set of data processors implicitly organized in a logical data flow transformation
- A defined data flow is run by a single executable - the DPL driver
- Includes a powerful GUI for logs/metrics and debugging
  - Especially helpful for individual users

# Analysis in RUN3:

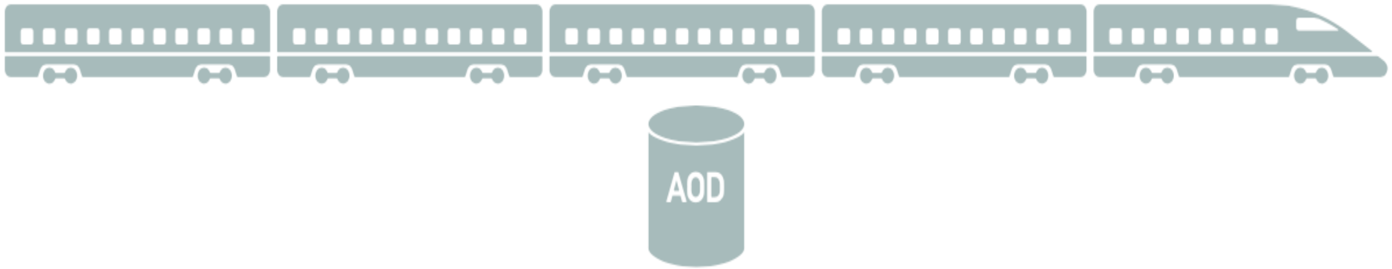
## Problem:

Analysis remains I/O bound in spite of attempts to make it more efficient by using the train approach

Data Analysis using ALICE Run3 Framework, 5 Nov, G.Eulisse, T6, 11:45

# Analysis in RUN3: (Solution)

- **Retain concepts that worked:** analysis trains, centralized code, abstraction framework
- Use better compression algorithms
- **Recompute** quantities on the fly rather than storing them.
- Flat data structures
- Only AODs for analysis



# Requirements for the AOD format

- AOD's data format will have to play well with AliceO2 message passing, shared memory backed, distributed nature.
- **Zero-{Copy, Serialisation, Adjustments}**:
  - *we want to be able to reuse data between processes.*
- **Growable**: *ability to extend columns on the fly.*
- **Prunable**: *ability to drop columns on the fly.*
- **Skimmable**: *ability to select only certain rows.*
- *Strategy: we are willing to lose some degree of generality for performance.*

# Apache Arrow

Data Analysis using ALICE Run3 Framework, 5 Nov, G.Eulisse, T6, 11:45

- Apache Arrow as backing store for the message passing.
- Arrow fits well to represent column oriented data, while providing some level of flexibility for nested data via the usual record shredding.
- Using Apache Arrow allows for seamless integration with a larger ecosystem of tools, like Pandas or Tensorflow.

# ALICE Analysis Facilities (Run3)

- Motivation
  - Analysis is the least efficient of all workloads that we run on the Grid
  - I/O bound in spite of attempts to make it more efficient by using the analysis trains
- Solution
  - Collect AODs on a few dedicated sites that are capable of locally processing quickly large data volume
  - Typically (a fraction of) HPC facility (20-30’000 cores) and 5-10 PB of disk on very performant file system
  - Run organized analysis on local data like we do today on the Grid

# Analysis facility @ GSI (Prototype for RUN3)

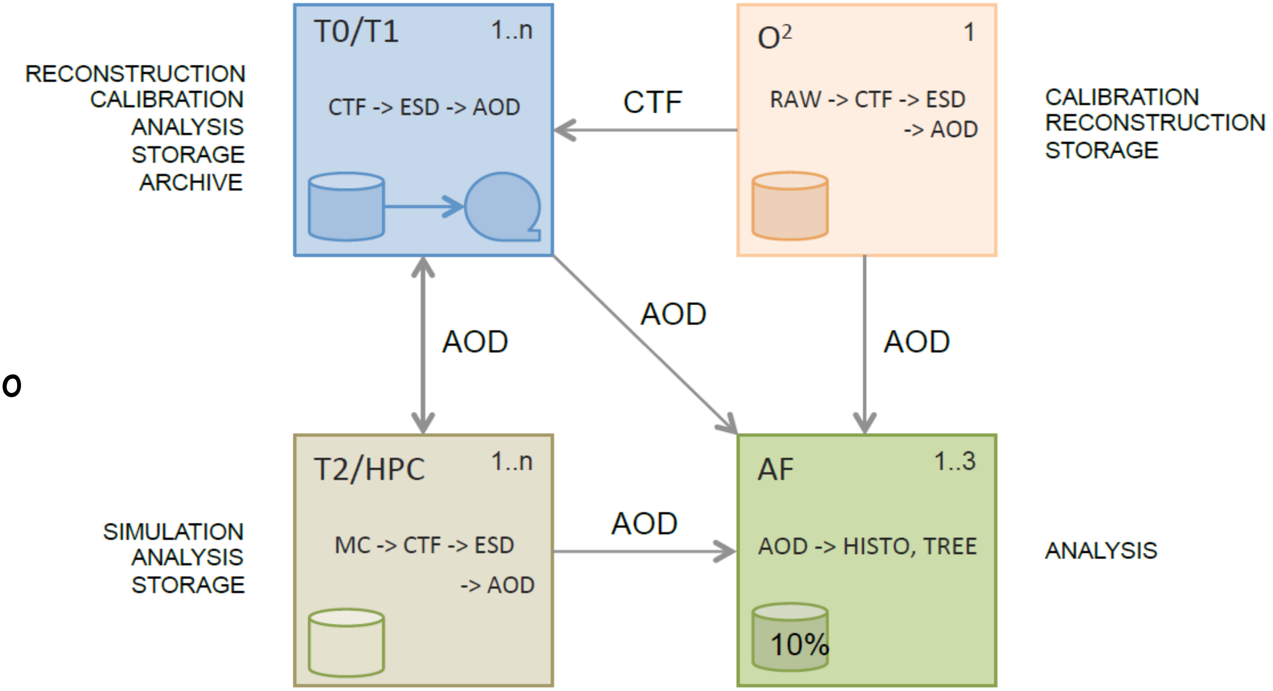
- GSI Tier 2 Resources
- Full AOD set 2015 Pb-Pb (LHC15o, about 250 TB)
- Shared file system (Lustre) + xrootd client plugin (0.6PB)
- Performance tests suggest that the target throughput rate of 10 PB/day can be achieved

A prototype for the ALICE Analysis Facility at GSI (2018)
https://indico.cern.ch/event/587955/contributions/2937941/

# Computing model in a single figure

Grid Tiers will be mostly specialized for given role

- O2 facility (2/3 of reconstruction and calibration),
- T1s (1/3 of reconstruction and calibration, archiving to tape),
- T2s (simulation)
- *All AODs will be collected on the specialized Analysis*

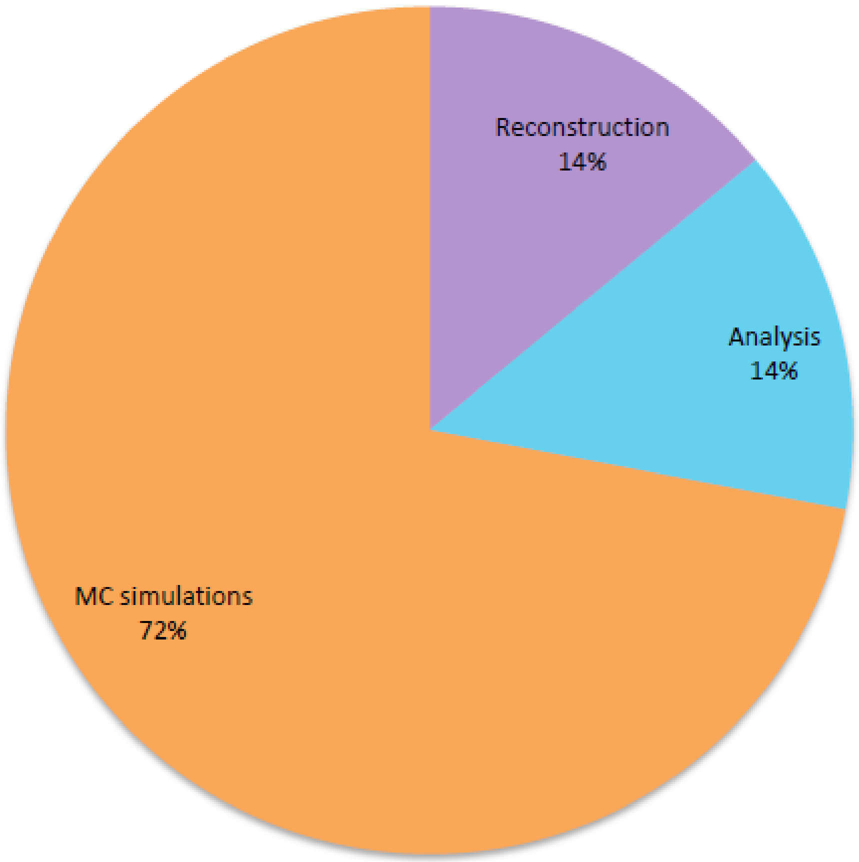


Facilities (AF) capable of processing ~5 PB of data within ½ day timescale

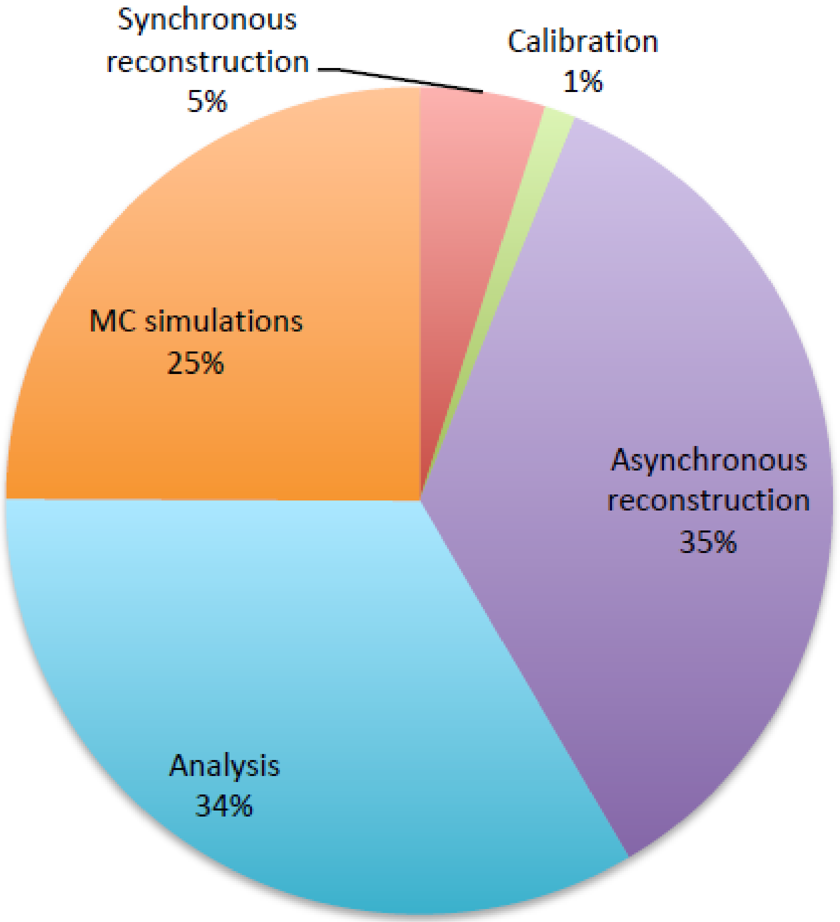
The goal is to minimize data movement and optimize processing efficiency

# Resources share projection

Run2



Run3+



# Summary

- Message Queues based solution (microservices) as a new paradigm for ALICE software
  - Different topologies of tasks can be adapted to the problem itself, and the hardware capabilities
- Apache Arrow as in memory backing store simplifies the interoperability with a number of OpenSource tools.
- Performance tests of the proto type AF at GSI, suggest that the target throughput rate of 10 PB/day can be achieved