INFN and The Future of Scientific Computing

Turin May 4, 2018

Using Hadoop ecosystem tools for distributed datacenters and the ALICE O2 farm monitoring

Gioacchino Vino (INFN Bari)

INFN BARI
Istituto Nazionale di Fisica Nucleare
Sezione di Bari

Tutor: Dott. Domenico Elia

Who am I?

- Postgraduate course in “Development and management of data centers for high performance scientific computing”, 2014-2015
  - Thesis title: “**Dashboard for the ALICE activity in Bari Tier-2 Site**”
  - Tutors: Domenico Elia and Antonio Franco

- Scholarship at GARR in “**Monitoring system for geographically distributed datacenters based on Openstack**”, 2016-2017
  - Tutors: Domenico Elia and Giacinto Donvito

- Scholarship at INFN, currently working on “**Monitoring of the ALICE O2 Facility**”, since Feb 2018
  - Tutor: Domenico Elia

Index

- Monitoring of geographically distributed datacenter based on OpenStack: MonGARR
  - Motivations
  - Project Overview
  - Future works
- Monitoring of the ALICE O2 Facility @CERN: Modular Stack
  - Architecture
  - Future works

MonGARR: Motivations

The increasing of computation resource demand for scientific purposes is leading to:

- Datacenters increasing in complexity and size.
- Taking advantages of new technologies like virtualization and cloud computing.
- Datacenter cooperation needed in order to accomplish common goals.

Geographically distributed datacenters

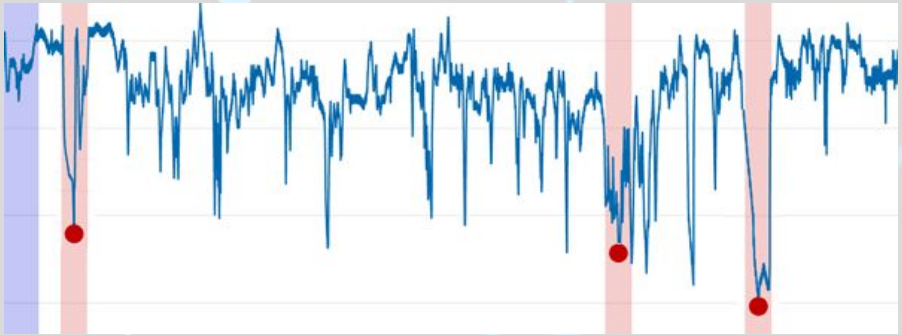
- Goal: Increase the computation capability of overall system.
- Side effect: Increasing complexity from the monitoring and control system.

Project: Developing a monitoring system for geographically distributed datacenters.

MonGARR: Project Overview

Advanced features are required:

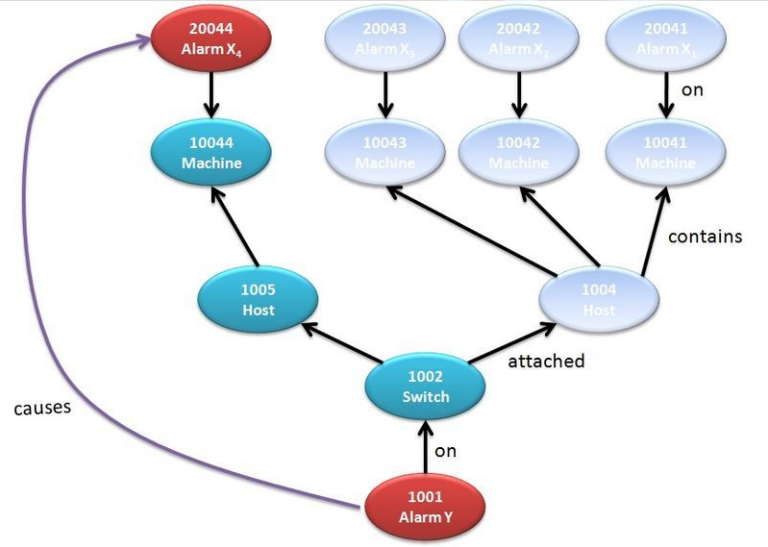
- Anomaly detector



MonGARR: Project Overview

Advanced features are required:

- Anomaly detector
- Root Cause Analysis



MonGARR: Project Overview

Advanced features are required:

- Anomaly detector
- Root Cause Analysis

Fully informative monitoring data are collected:

- Service monitoring (HTTP server, DBs, …)
- Openstack and middleware monitoring
- Hardware monitoring (physical servers, disks, disk controllers, network devices, PDU, …)

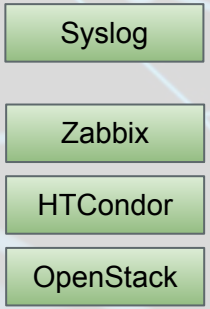
MonGARR: Project Overview

Testbed

ReCaS Bari Datacenter:

- More than 13.000 cores
- 7.1 PB Disk Storage
- 2.5 PB Tape storage
- HPC Cluster composed of 20 servers
- Dedicated network link: 10Gbps x2 to GARR, 20Gbps to Naples and 20 Gbps to Bologna
- Cloud platform: OpenStack
- Batch system: HTCondor
  - 184 Worker Nodes
  - 350+ network connections
- Local Monitoring System: Zabbix
- Including ALICE and CMS Tier2s

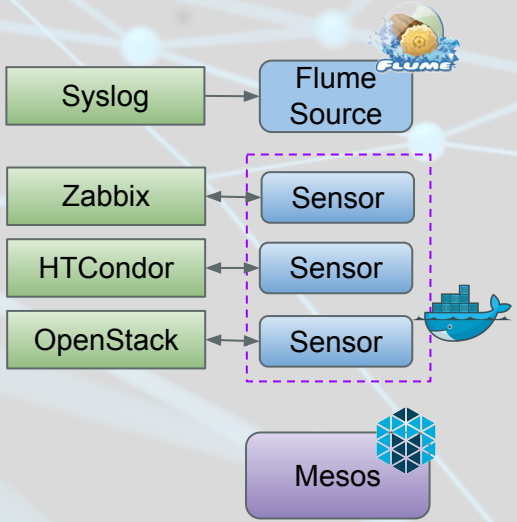
MonGARR: Project Overview



Data Sources:

- **Syslog**: System processes and service information.
- **Zabbix**: Computation resource usage, service and Openstack monitoring.
- **HTCondor**: Scheduler, completed and running job state
- **OpenStack**: Information on server, images, flavors, volumes, network devices, ….

MonGARR: Project Overview

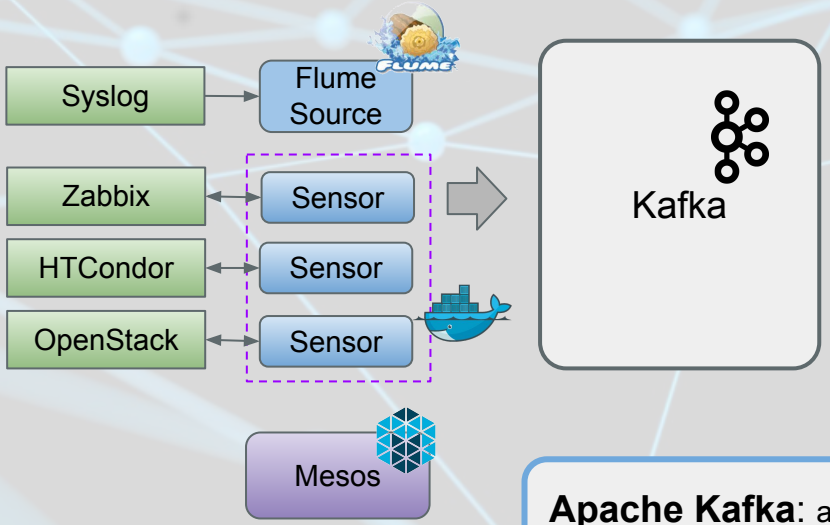


Metric collectors:

- **Apache Flume** Syslog Source.
- Python code inserted in **Docker**-container and executed periodically using **Apache Mesos.**

Apache Flume: a distributed and highly-reliable service for collecting, aggregating and moving large amounts of data in a very efficient way.
Apache Mesos: an open-source project to manage computer clusters.
Docker: a computer program that performs operating-system-level virtualization also known as containerization.

MonGARR: Project Overview

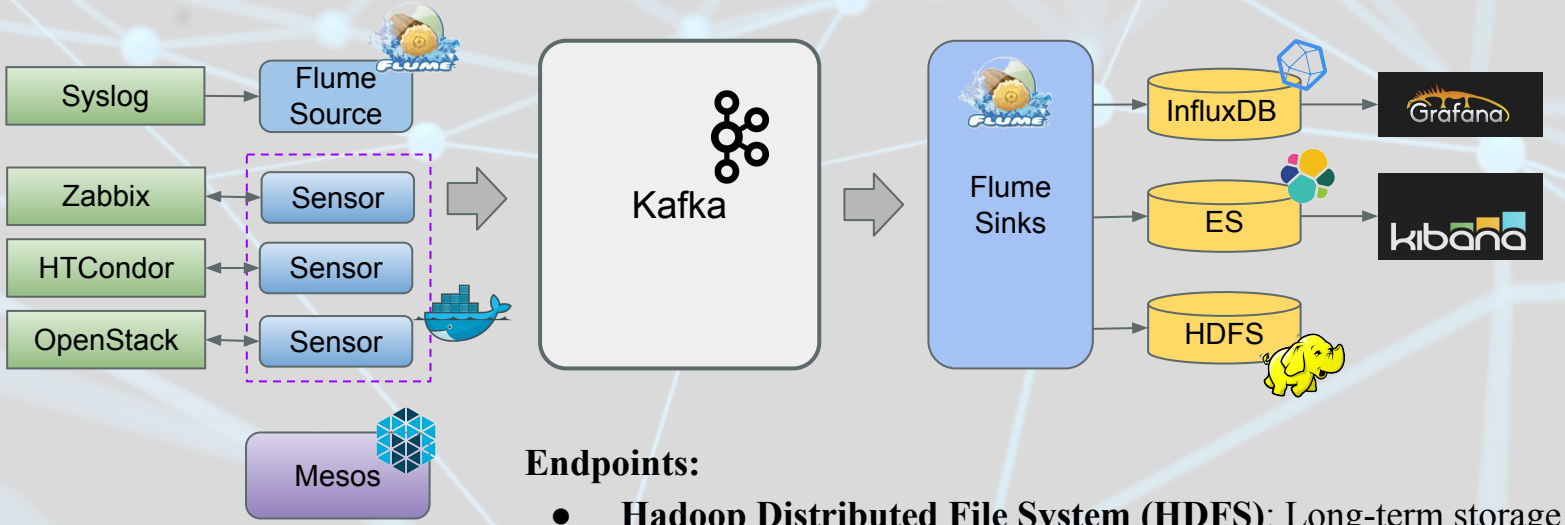


Transport Layer:

- **Apache Kafka.**
- Decouple all components.
- Increase the High Availability of system.

Apache Kafka: an open-source stream-processing software platform, provides a unified, high-throughput, low-latency platform for handling real-time data feeds.

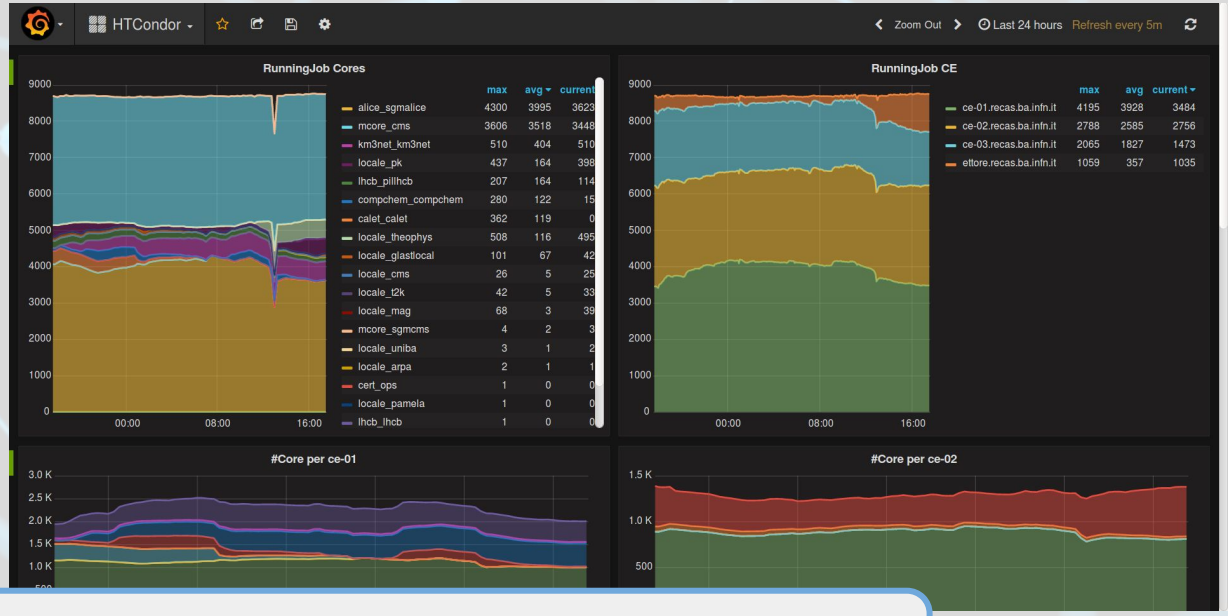
MonGARR: Project Overview



Endpoints:

- **Hadoop Distributed File System (HDFS)**: Long-term storage.
- **InfluxDB-Grafana**: Timeseries Dashboards.
- **ElasticSearch-Kibana**: Log Dashboards.

MonGARR: Project Overview

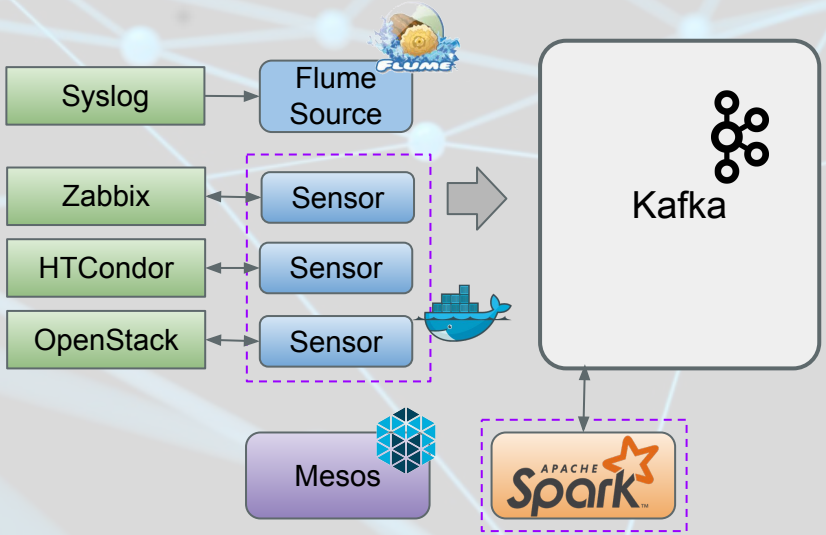


InfluxDB: a custom high-performance data store written specifically for time series data.
Grafana: Dashboards' builder for time-series data.

MonGARR: Project Overview

ElasticSearch: a search engine based on Lucene and provides a distributed, multitenant-capable full-text search engine with an HTTP web interface and schema-free JSON documents.
Kibana: an open source data visualization plugin for Elasticsearch

MonGARR: Project Overview

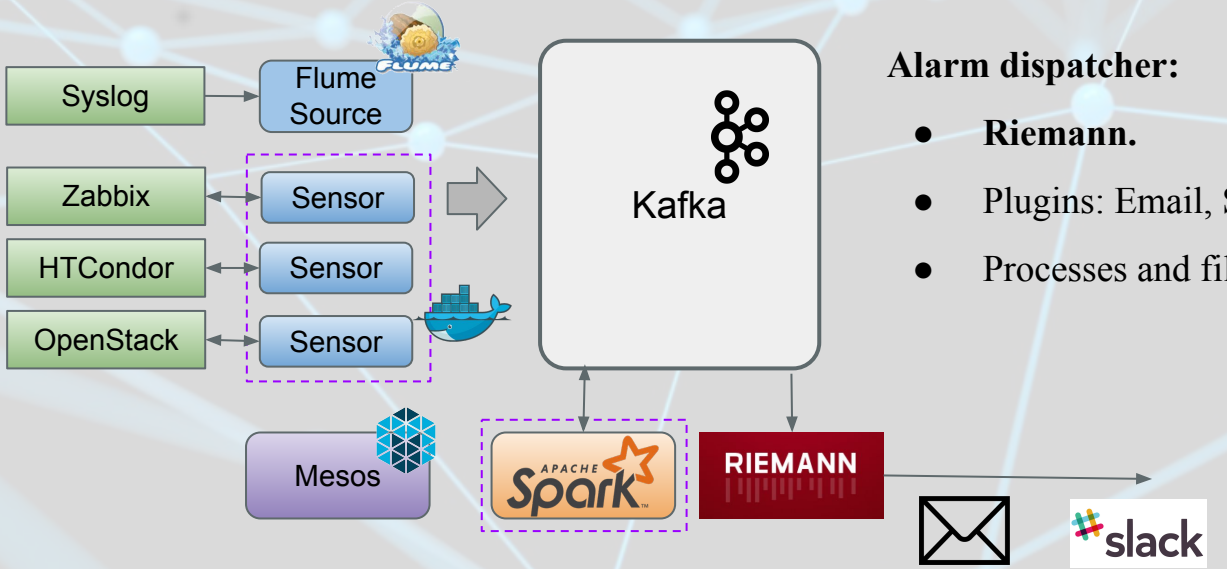


Processing Unit:

- **Apache Spark.**
- Log Analyzer.
- Anomaly Detector.
- Data Correlation.
- Root Cause Analysis.

Apache Spark: a fast and general engine for large-scale data processing.

MonGARR: Project Overview

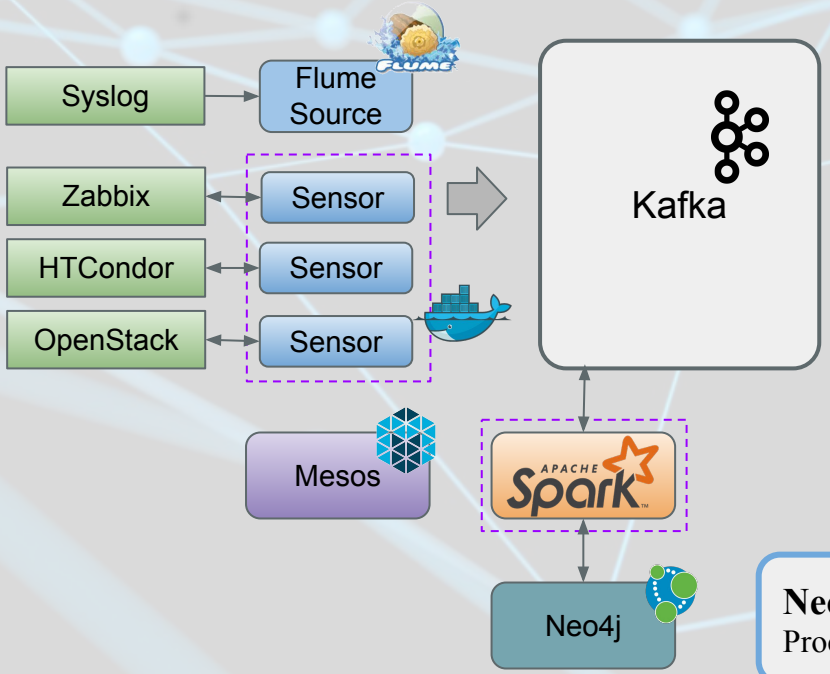


Alarm dispatcher:

- **Riemann.**
- Plugins: Email, Slack.
- Processes and filters events.

Riemann: aggregates events from your servers and applications with a powerful stream processing language.

MonGARR: Project Overview



Information Structure:

- Classical monitoring is not enough.
- Relation information (Services, network, virtual-physical server, …)
  - Openstack data.
  - Open connections.
  - Other monitoring data.

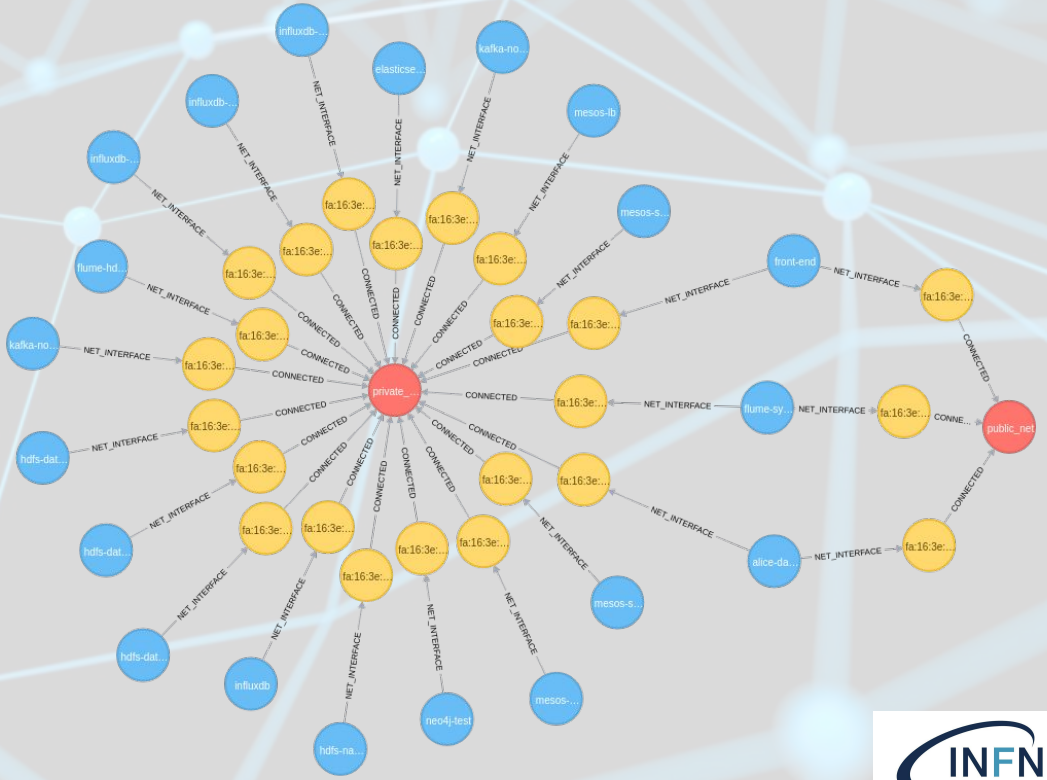
Neo4j: High Performance native Graph Storage & Processing.

MonGARR: Project Overview

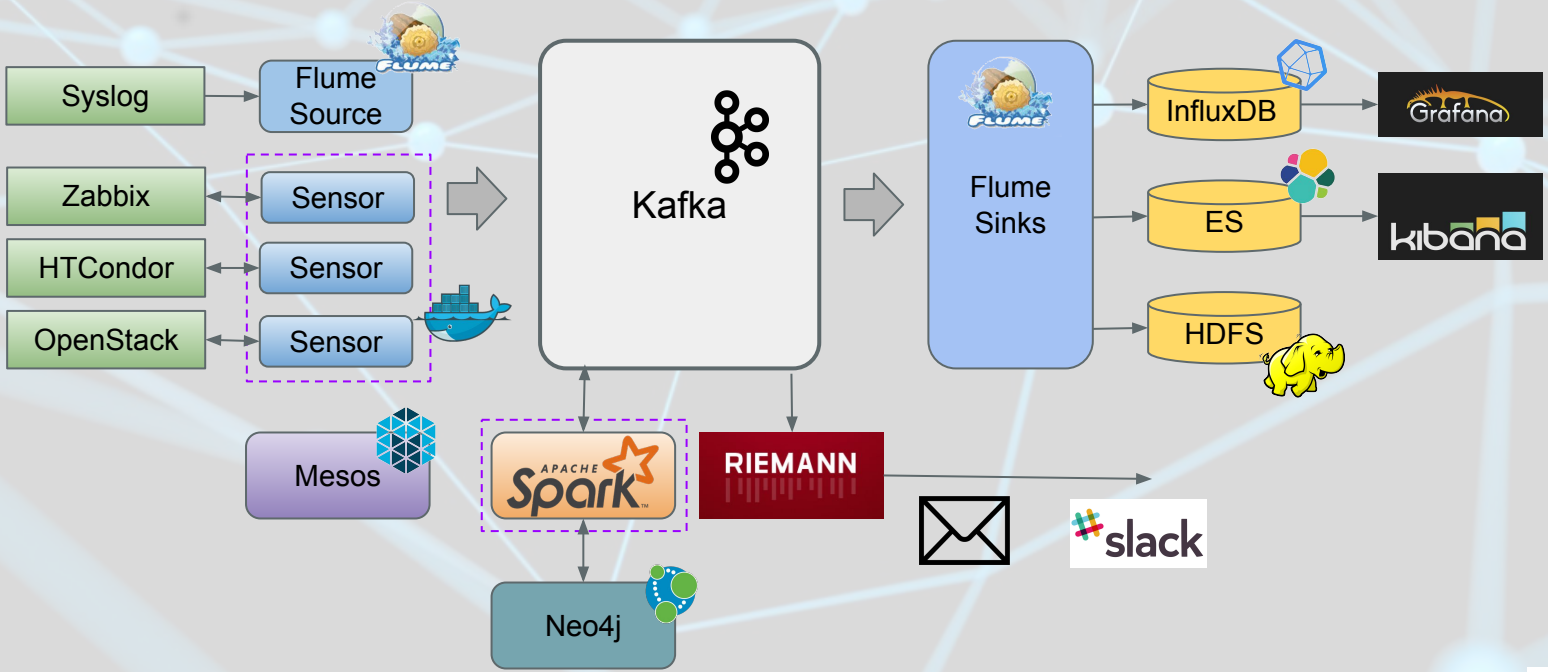
Information Structure:

Subgraph example:

- Blues nodes: virtual machines.
- Yellow nodes: network interfaces.
- Red nodes: networks.



MonGARR: Project Overview



MonGARR: Project Overview

Resource Usage for the monitoring system:

- 80 CPUs
- 150GB RAM
- 3 TB Disk
  - 1.5TB for HDFS in replica 3
  - 600 GB for Kafka nodes
  - No-volatile virtual machine volumes

MonGARR: Project Overview

Apache Mesos:

Cluster:

- 3x Master (2 CPUs, 4GB RAM, 20GB Disk)
- 2x Slaves (4 CPUs, 8GB RAM, 20 GB Disk)
- 1x Load Balancer (2 CPUs, 4GB RAM, 20GB Disk)

Frameworks:

- Chronos
- Marathon
- Spark

MonGARR: Future works

- Migrate all components in Mesos
- Improve the Machine Learning algorithms efficacy
- Root Cause Analysis algorithm
- Integration with project management systems (OpenProject, Trello, ….)

Modular Stack solution for ALICE O2 monitoring

- ALICE is a heavy-ion detector designed to study the physics of strongly interacting matter (the Quark–Gluon Plasma) at the CERN Large Hadron Collider (LHC).
- During the Long Shutdown 2 in the end of 2018, ALICE will start its upgrade to fully exploit the increase in luminosity.
- The current computer system (Data Acquisition, High-Level Trigger and Offline) will be replaced by a single, common O2 (Online-Offline) system.
- Some detectors will be read out continuously, without physics triggers.
- O2 Facility will compress the 3.4 TB/s of raw data to 100 GB/s of reconstructed data

- Development of a Monitoring System for ALICE O2 Facility:
  Modular Stack solution, with components and tools already used and tested in the MonGARR project (approved by the ALICE O2 TB last February)

Modular Stack solution for ALICE O2 monitoring

ALICE O2 Facility:

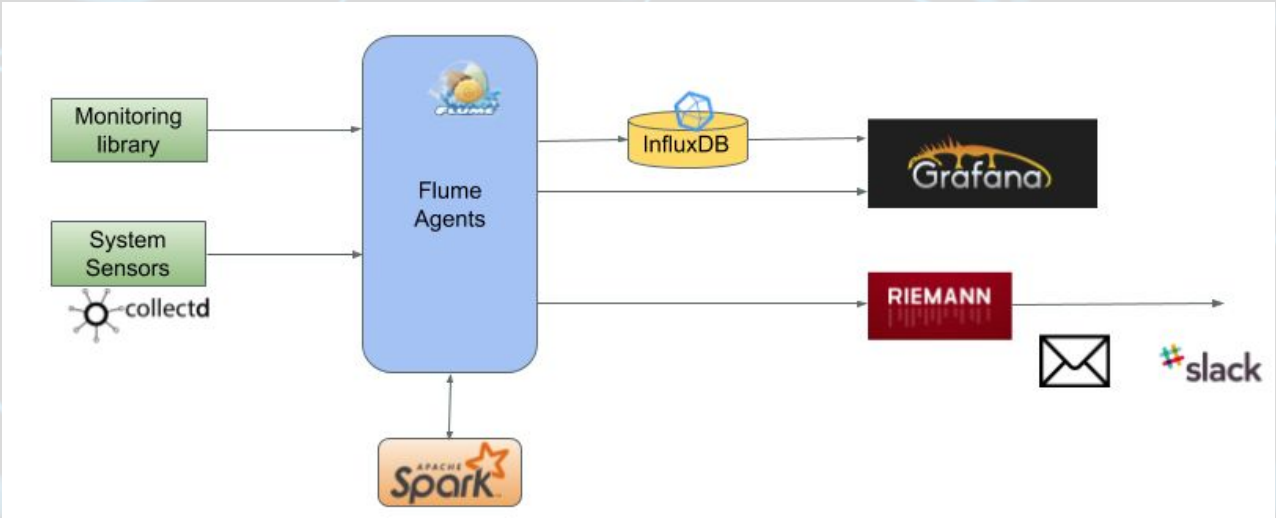
- 268 First Level Processors
- 1500 Event Processing Nodes

Requirements:

- Capable of handling O2 monitoring traffic – 600 kHz
- Scalable >> 600 kHz
- Low latency
- Compatible with CentOS 7
- Open Source, well documented, actively maintained and supported by developers
- Impose low storage size per measurement

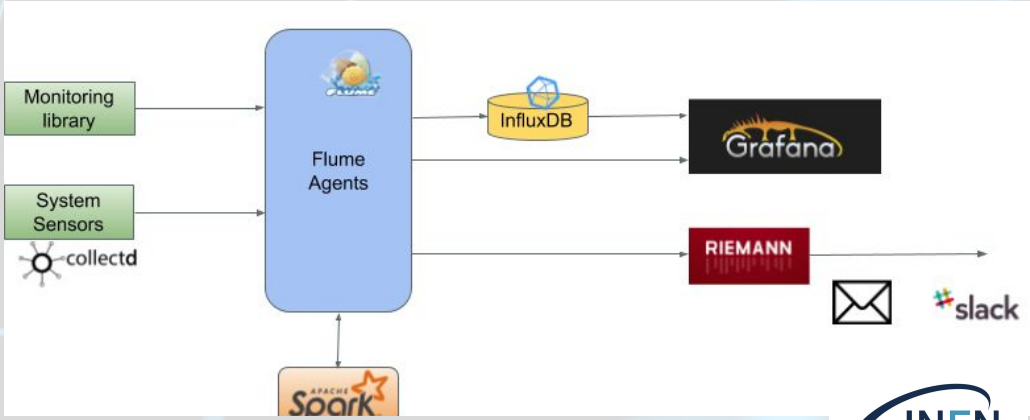
Modular Stack: Architecture

- Sensors:
  - **Monitoring Library**
  - **CollectD**
- Transport Layer:
  - **Apache Flume**
- Time-series Database:
  - **InfluxDB**
- Visualization interface:
  - **Grafana**
- Alarming component:
  - **Riemann**
- Processing component:
  - **Apache Spark**



Modular Stack: Architecture

- Sensors:
  - **Monitoring Library:** user defined metrics, monitoring process metrics
  - **CollectD:** CPU, network, memory, load, uptime, disk, log files,....
- Transport Layer:
  - **Apache Flume**: implemented custom components
- Time-series Database:
  - **InfluxDB**
- Visualization interface:
  - **Grafana:** users, teams, dashboard
- Alarming component:
  - **Riemann**: Slack alarm
- Processing component:
  - **Apache Spark:** aggregation jobs



Modular Stack: Future works

- I System Validation using the TPC monitoring data, May 2018
- New functionalities will be added (new streaming analysis, alarming, log analysis)
- II System Validation using ITS monitoring data, Dec 2018

THANKS FOR

YOUR

ATTENTION

INFN and the Future of Scientific Computing
4-may-2018

DPM-based distributed caching system for multi-site storage in ATLAS

Bernardino Spisso
INFN Napoli

About me

- Master degree in theoretical physics at Federico II di Napoli (2007).
- Ph.D. in mathematics at Westfälische Wilhelms-Universität of Münster (2012).
- I level University Master Course in “Technologies for high-performance scientific computing” at Federico II di Napoli (2014).

My previous occupation was in the ASTERICS-Km3Net collaboration, responsible for the projects:

- ROAst (ROOT extension with ASTrophysical).
- CORELib (COsmic Ray Event Library).

Since December 2017 I collaborate with Naples ATLAS computing Group:

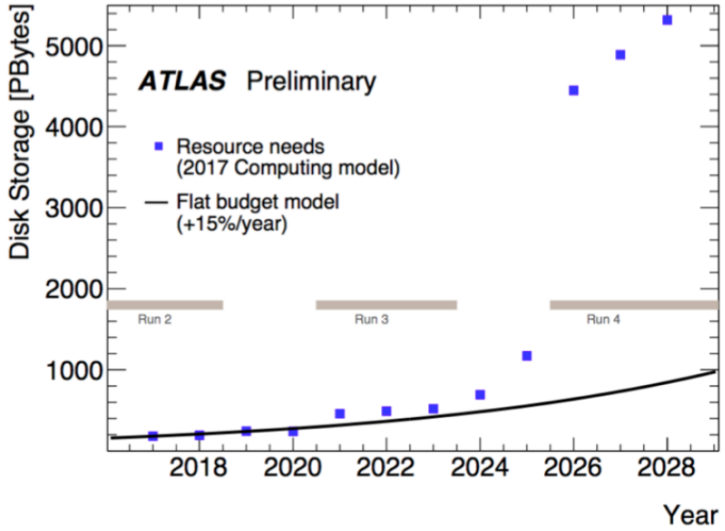
- Alessandra Doria and Giampaolo Carlino.

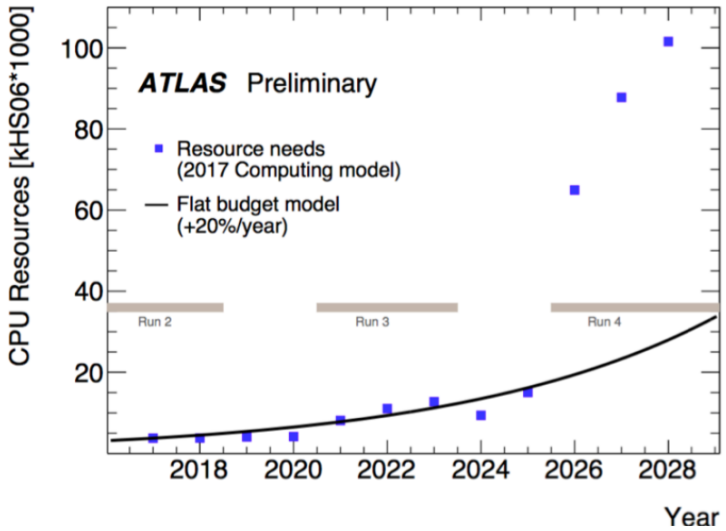
And with the ATLAS Tiers-2 at LNF and ROMA1.

- Elisabetta Vilucchi (INFN-LNF) and Alessandro De Salvo (INFN-Roma1).

Motivations

- HL-LHC storage needs are above the expected technology evolution (15%/yr) and funding (flat).
- We need to optimize storage hardware usage and operational costs.





Plots from the last Joint WLCG-HSF Workshop 2018 in Naples.

Our scenario

- Explore distributed storage evolution to improve overall costs (storage and ops) taking in account:
  - Single common namespace and interoperability.
  - User analysis is often based on clusters hosted on medium sites (Tier2) and small sites (Tier3).
- **In order to reconcile these two trends, the target of my activity is to study a distributed storage system featuring a single access point to large permanent storage and capable to provide efficient and dynamic access to the data. In this view, medium sites like Tier2 and small sites like Tier3 will not necessarily require large storage systems, simplifying local management**.

This can be achieved by the adoption of a distributed storage and caching technologies.

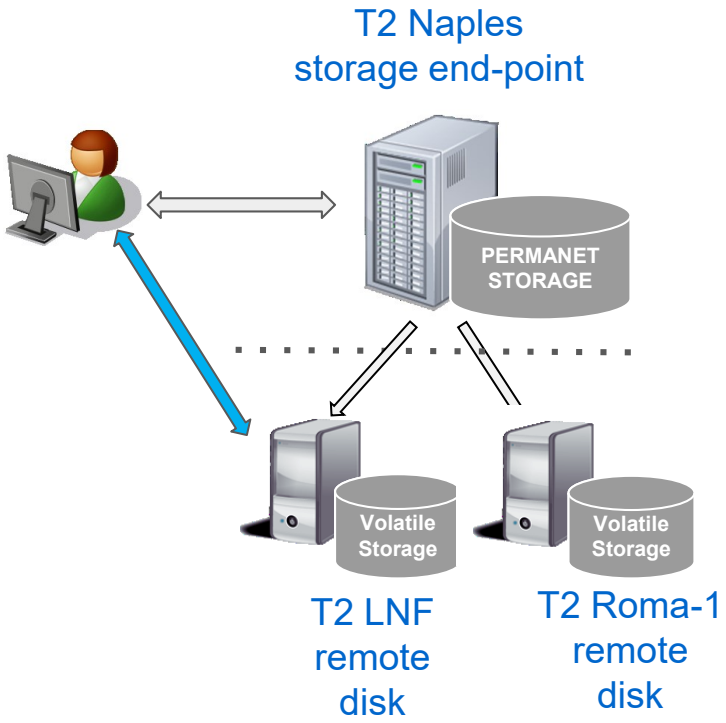
- This activity takes place in the same context of the Data Lake project having very similar motivations.

Our implementation

- The Disk Pool Manager (DPM) is a data management solution widely used within ATLAS, in particular in three Italian Tier2.
- The latest versions of DPM are used in our implementation, that offer the possibility to manage volatiles pools to be used as caches.

By exploiting the fast connections between sites, we are deploying a first testbed among Naples, Frascati and Roma-1 using DPM. The aim is to study and develop a configuration in which a primary site represents a single entry point for the entire archiving system and each site can use its storage as permanent storage or as local cache.

Using a cache system the local site administrators can be dispensed from managing a complete storage system. The site became transparent for the central operations of the experiment.



Conclusions

- A first testbed using DPM among Naples, Frascati and Roma-1 is almost ready.
- Study of the best caching policy for the volatile pools.
- Evaluation of the performance of the developed prototype.
- System integration in the current ATLAS data management infrastructures.
- Synergies:
    - collaborations with the Naples BELLEII computing group (Silvio pardi (INFN-NA), Davide Michelino (GARR))
    - collaborations with the DPM development group.
- Create conditions for easy replication of the system on other sites or in other contexts.

Miguel Villaplana

- ATLAS - Milano
- Supervisor: Laura Perini

ATLAS EventIndex **(in collaboration with Dario Barberis)**

- **A database with the references to the files including each event in every stage of processing**
  - fast and efficient selection of events of interest, based on various criteria, from the billions of events recorded
  - an indexing system that points to those events in millions of files scattered in a world-wide distributed computing system
  - contains records of all events processed by ATLAS, in all processing stages

- **Contribution to Functional Tests and User support**
  - Based on previous work from A. Favareto: bash script
  - Redesigned from scratch using python
    - modular design, exception handling, code documentation
  - Added new functionality
    - test MC samples
    - resubmission of failed tests

- **Status and plans**
  - Short term: machinery is mostly in place but needs polishing
  - Long term: design and implementation of the functional tests for the Event Whiteboard (EI's evolution)

Harvester (in collaboration with Alessandro De Salvo)

- The current production and distributed analysis system in ATLAS (PanDA) relies on a server-pilot paradigm
  - A server maintains state and manages workflows with various granularities
  - Pilots are job-centric and run independently on worker nodes with a limited view of local resources

- PanDA itself has no means of managing and monitoring cloud utilisation

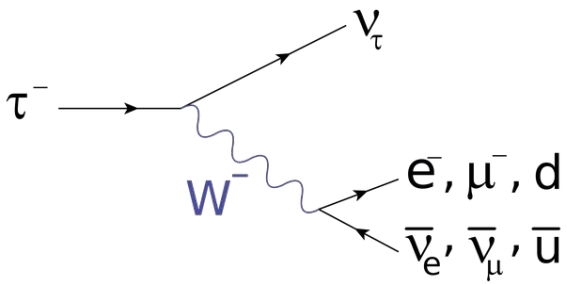
- **Harvester is a resource-facing service between the PanDA server and the collection of pilots**
  - It is a stateless service with knowledge of the resources
  - It can act as an intermediate communication channel between PanDA server and pilots

- **Current activity**:
  - Our goal is to improve site deep resource knowledge
  - Start by using the available information sources (GLUE 1.2, GLUE 2)
    - Understand if the available information sources could be reliable and useful w.r.t. what we currently have in Panda
      - Compare the values obtained from Glue 2 and what we have in Panda
      - Many problems found: misconfigured sites, only a fraction (< 40%) of sites/CEs have matching values between Glue2 and Panda
    - In summary, GLUE values do not really seem reliable enough

- **Future plans:** grabbing values directly from jobs
  - Initial prototype of a collector designed by A. De Salvo
  - Plan to contribute to the development and testing of the prototype

Machine learning

- **H -> Tau Tau mass reconstruction (in collaboration with Attilio Andreazza)**



  - Tesi di laurea di Aldo Materassi

  - **Goal**: predicting the invariant mass of ditau system using the visible tau decay products kinematics
    - Difficult final state: many neutrinos

  - **Status and plans:**
    - Now learning basics about ATLAS software and how to get/process the data
    - We will first reproduce previous results
      - using BDTs (and random forests)
      - can we train at truth level and test at reco level?
      - best way to use ML frameworks in the market (scikit-learn, pyTorch,...) inside ATLAS software
    - Try other ML techniques

Share resources across groups at UniMi

- **Groups are encouraged to organise their resources under HTCondor pools**
  - Execute machines report to the central manager of their own pool

- **We add an additional central manager to which all execute machines report too**
  - This provides usage accounting across all the resources together
  - Serves as a top-level pool to submit jobs to when users want to access all possible resources

- **Users get the quality of service they were already enjoying, but excess jobs may be conveniently sent to the other resources**
  - Group pools remain the default pool for job submission, but with the super-pool added to their FLOCK_TO list
  - We give the group's negotiator priority over super-pool's to guarantee high priority to group users on their own machines

- **Status and plans**
  - The system is in place and shows good behaviour
  - ATLAS use-case:
    - Trivially parallel jobs. CVMFS is the only requirement.
  - Currently researching how to run MPI jobs encapsulated in Docker containers via HTCondor
    - Tesi di laurea triennale di Massimo Miserendino (sup. F. Prelz, D. Rebatto and A. Andreazza) http://infn.it/thesis/thesis_dettaglio.php?tid=12211
  - Monitoring and accounting
    - Based on Filebeat+Elasticsearch+Kibana
  - Documentation and user support
    - Scripts and conf files in GitLab @ INFN (baltig.infn.it)
    - Instructions for sys. andmins growing both in GitLab and in a local twiki
    - User support: twiki + mailing lists for support and announcements in construction

Machine Learning as a Service

(for the ALICE collaboration)

Sara Vallero

INFN Torino

Introducing myself

Within the ALICE collaboration

- Data analysis
- Detector testing/commissioning
- Data-taking operations

Building competences

- Machine Learning (ML) algorithms
- Distributed ML systems
- Integration of non standard resources (GPUs) with Linux containers and orchestrators (Mesos, OpenNebula)

PRIN Project
(optimization of access to LHC data using the grid and cloud computing)

2013

INDIGO-DC

2015

THIS FELLOWSHIP

2007
MD in Physics
(University of Torino)

2012
PhD in Physics
(University of Heidelberg)

March 1, 2018

Cloud Computing

(full virtualization)

- OpenNebula and KVM
- Contextualization of complex services
- Auto-scaling
- FairShare scheduling

Computing Model as a Service

(lightweight virtualization)

- runtime/application packetization (Docker)
- Distributed scheduling (Mesos and its frameworks)
- Network virtualization (Calico)
- HPC: MPI over InfiniBand in Docker

Batch System as a Service employed at the INFN and UNITO's HPC Cluster (OCCAM)

Machine Learning as a Service

Goals
Architecture
Technologies
Implementation
Test use-case

CHALLENGES

- Reconstruction
- Analysis
- Trigger
- Data quality
- Detector monitoring
- Computing operations
- Monte Carlo tuning
- …

REQUIREMENTS

- Workflow definition
  - Results reproducibility
- Multi-tenancy (scheduling, authentication…)
- Parallel execution and scaling
- Data handling
- Ease of use and management
- …

IMPLEMENTATION

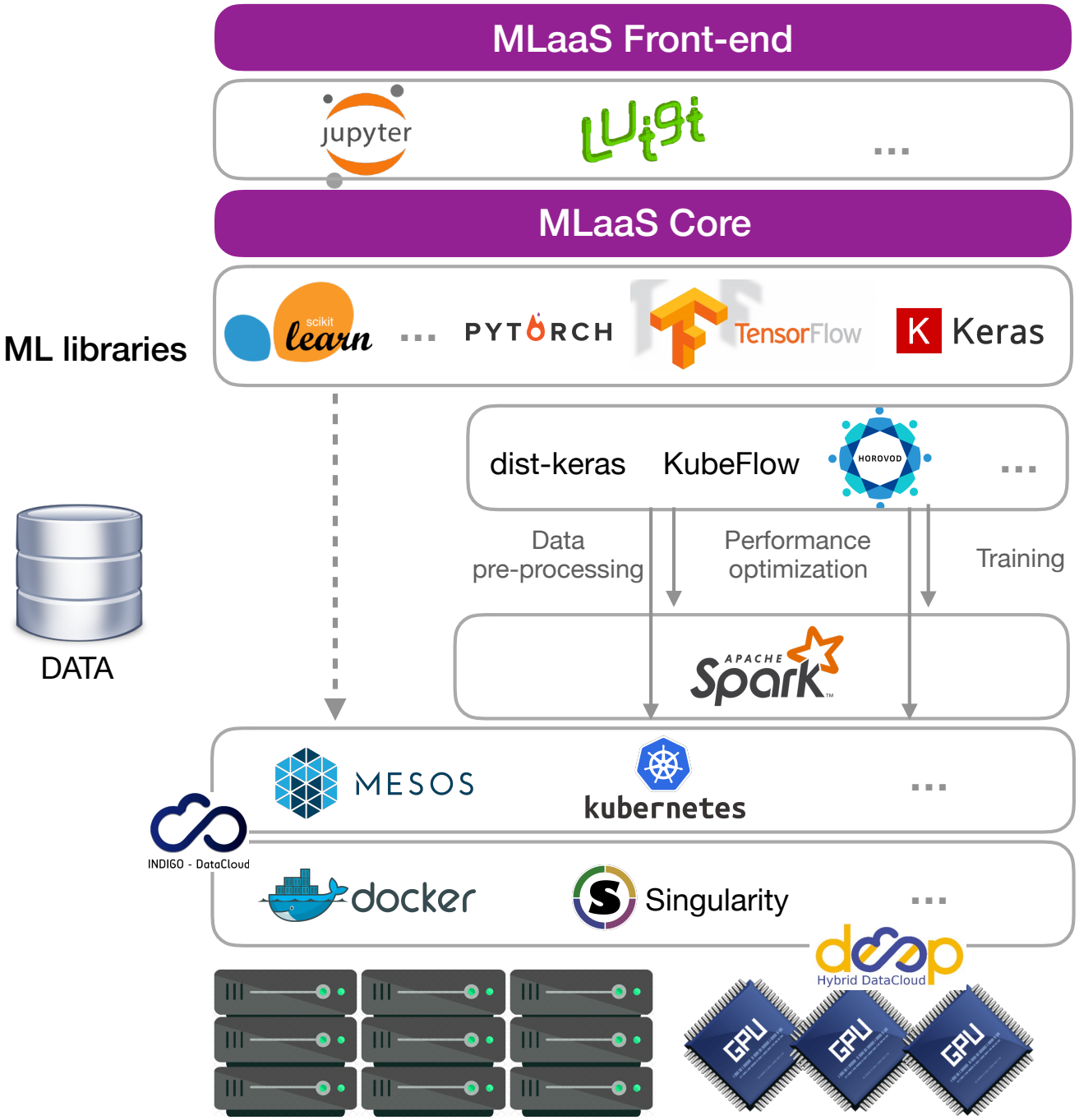
- Lightweight virtualization
- Modularity
- Flexibility
- Heterogeneous back-end infrastructures
- …

LEVERAGING

- Existing OpenSource software (mature and maintained)
- INDIGO-DataCloud products
- DEEP Hybrid DataCloud products

Brainstorming

Test use-case
Implementation
Technologies
Architecture
Goals



- Workflow definition
- Process Monitoring
- Authentication

INDIGO - DataCloud

Deep Learning framework

Distributed DL libraries

Cluster framework (parallelize task)

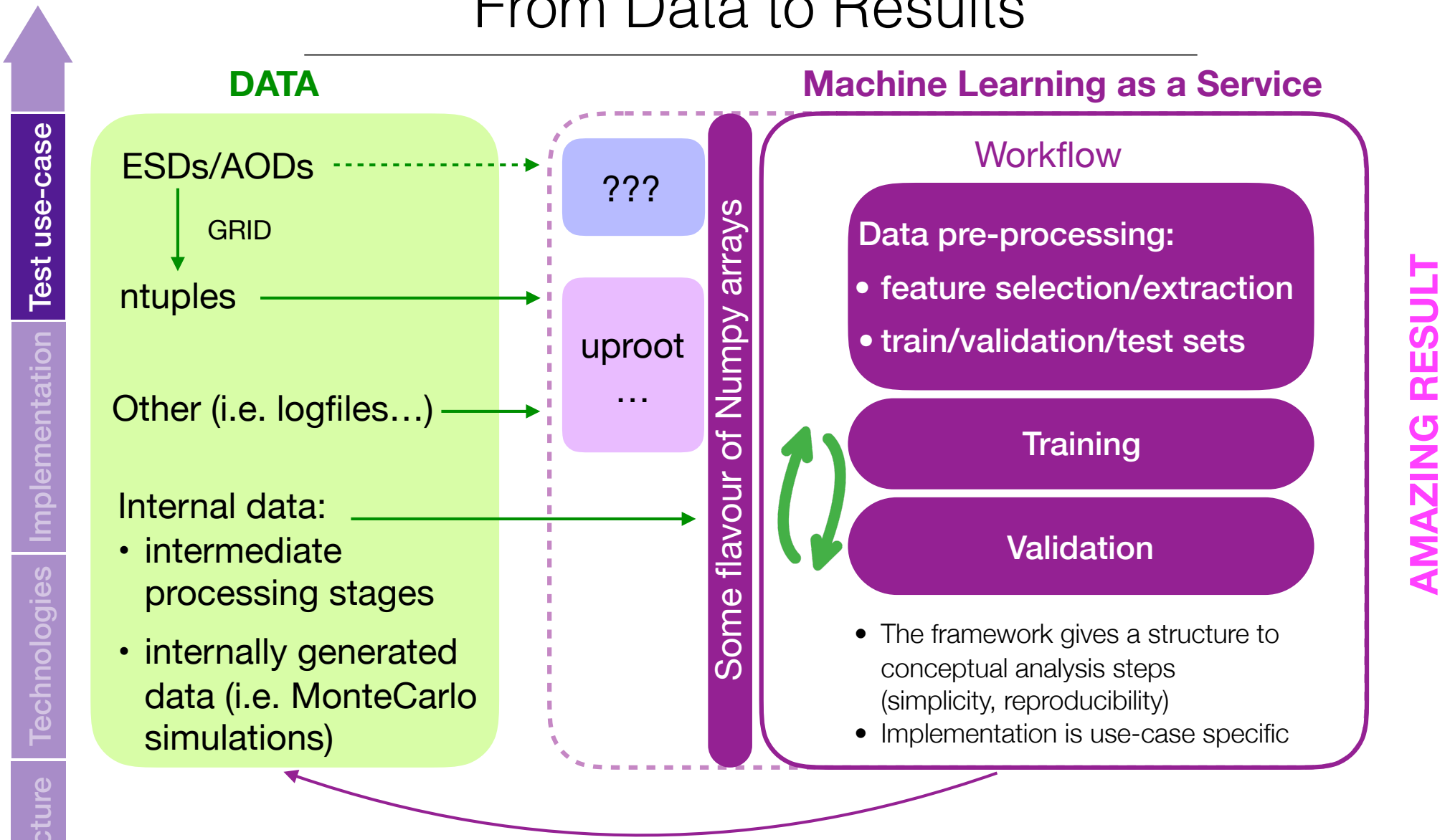
Orchestrator (schedule on resources)

Packetization and virtualization

Resources:

- Bare metal
- Infrastructure as a Service

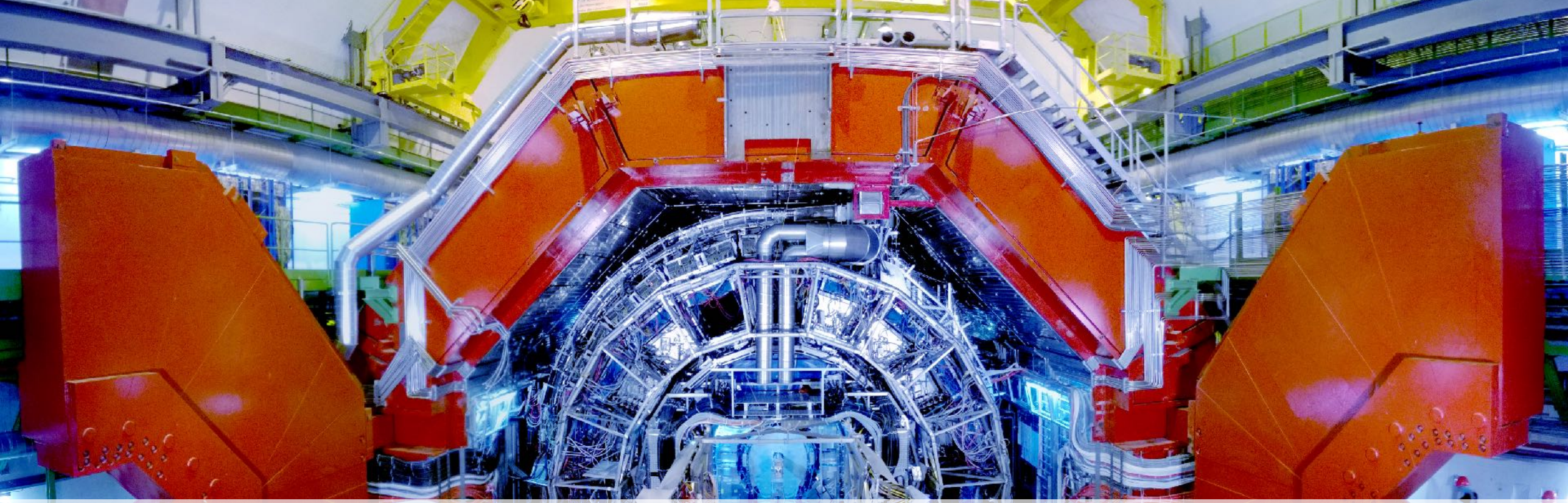
From Data to Results



Case study (ambitious): framework for systematic tuning of MonteCarlo generators

- data-MC comparison with Neural Network based high-dimensional discrimination
- learn event re-weighting function to avoid several expensive generation calls
- tune generator parameters by back-propagation

From M. Paganini's talk at the 2nd IML Workshop, April 9-12 2018, CERN



ALICE Analysis in Run 3

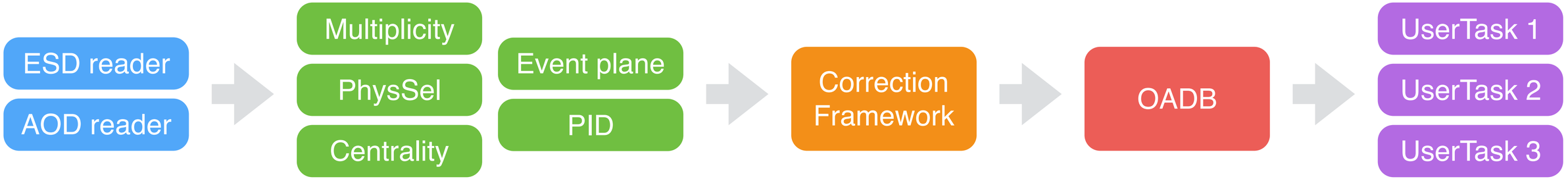
Dario Berzano

CERN and INFN Torino

Overview of the current ALICE analysis framework

- **Abstraction layer:** “Analysis Tasks” per-event processing function in ROOT/C++
- **Platforms:** same analysis can run locally, on Grid and ROOT’s PROOF
- **Data format:** reading from reconstructed (ESD) or analysis (AOD) data (ROOT TTrees)
- **Extensibility:** ancillary deltaAODs and ESD friends
- **Grid processing rate:** 5 MB/s of input data processed on each Grid job
- **Operations:** organized trains: read event once, process with many analysis tasks

Analysis data flow



Motivation for a new framework for Run 3

- **A new experiment:** trigger-less, no “events”, no 1:1 mapping to current analysis tasks
- **Non-linear analysis flow:** allow for more complex workflow connections
- **Insufficient rate:** 100x raw data expected, **how to keep up to it?**
  - Declarative approach: users don’t code the full workflow but connect custom or common task units, allowing the framework for an easier optimization
  - Optimize costly data format operations (serialization, compression, cross-referencing)
  - Read from local files bypassing central file catalog operations
  - Do not run on the Grid, but on specially designed HPC facilities with fast network/disk
  - Rethink analysis flow: instead of producing the final results immediately on HPC resources, do a coarse-grained selection for easier on-laptop optimization iterations

Development areas

Data format

Low deserialization cost

Efficient in-memory store

Optimized decompression

Analysis facilities

Only analyze local data

Fast local storage and network

Allow inter-nodes communication

Workflow handling

Allow for non-linear workflows

Nodes subscribe to data

Use network and shared memory

User interface

Reuse standard interfaces

Declarative paradigm

Optimize common operations

Large scale operations

Interface to central File Catalog

Compose analyses into trains

Test and monitor user jobs

Development areas

Data format

Low deserialization cost

Efficient in-memory store

Optimized decompression

Analysis facilities

Only analyze local data

Fast local storage and network

Allow inter-nodes communication

Workflow handling

Allow for non-linear workflows

Nodes subscribe to data

Use network and shared memory

User interface

Reuse standard interfaces

Declarative paradigm

Optimize common operations

Large scale operations

Interface to central File Catalog

Compose analyses into trains

Test and monitor user jobs

Data format

- **Base processing unit:** timeframes (not events: no trigger), 20 ms long (so quite long)
- **Flat and simple:** store numbers only, also for cross-referencing indices between tables
- **Columnar:** represent chunks of records as SoA in memory to leverage vectorization
- **Zero size for null objects:** filtered-out fields do not use RAM memory
- **Extensibility:** base format will never change, but easily extensible because it's SoA
- **Computing vs. storing:** in some cases, recomputing some fields is cheaper than storing
- **No data restructuring:** disk → memory → network use the same representation

Apache Arrow

LibFlatArray

SOAContainer

Data format implementations

SoA: good for vectorization but counter-intuitive: we think event-by-event (AoS). Lazy data access interfaces/formats to write AoS code transparently executed vectorized

- **Apache Arrow** - https://arrow.apache.org/
  Developed for analytics. Standard exchange format in the Apache Foundation ecosystem. Easy to organize and cross-reference data types with tables. Parquet as file backend

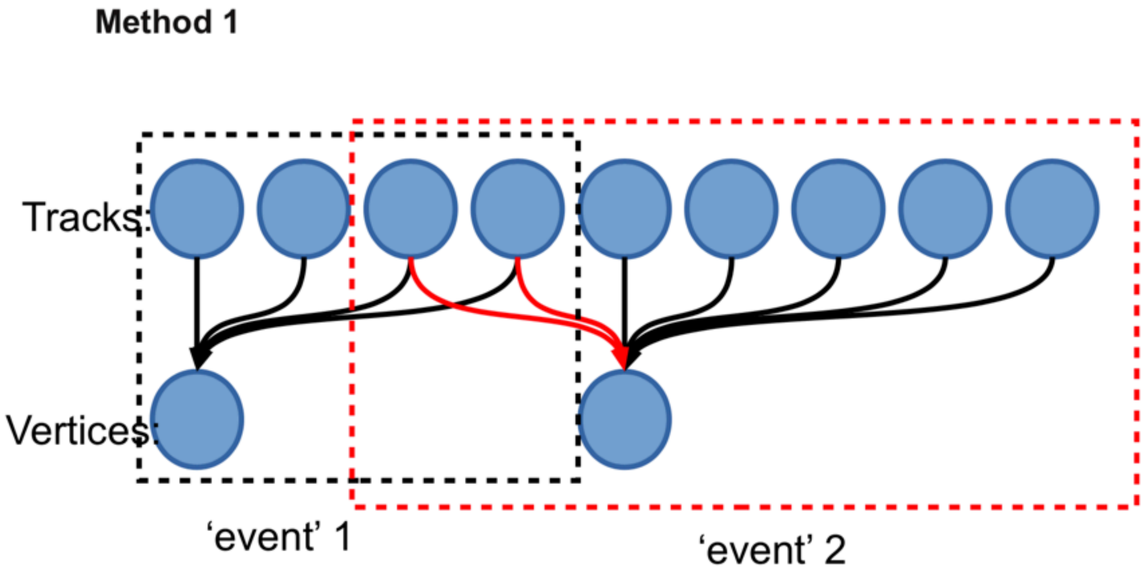
- **LibFlatArray** - http://www.libgeodecomp.org/libflatarray.html
  Developed for scientific applications. Provides an object-oriented interface to SoA. Leverages code generation through templates

- **SOAContainer** - https://gitlab.cern.ch/LHCbOpt/SOAContainer
  Developed by LHCb, currently in use for some HLT operations. Provides a std::vector-like container for looping easily over SoA collections

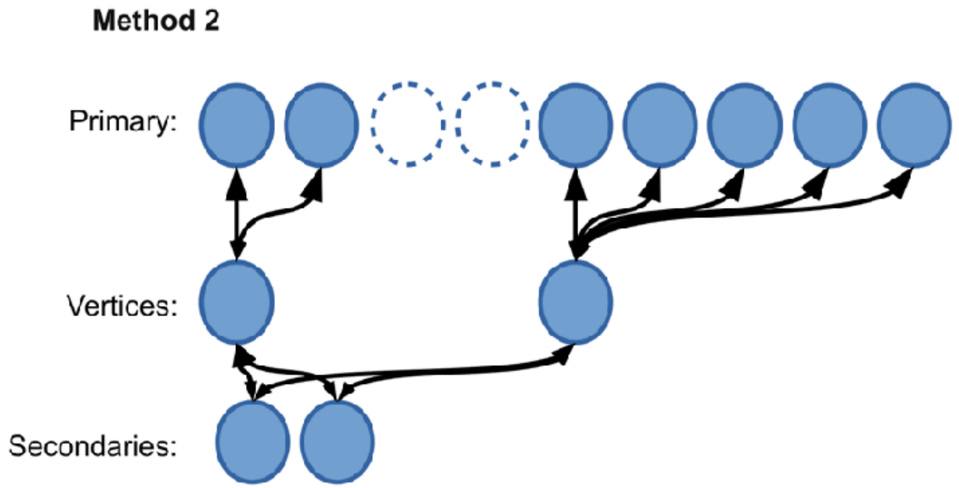
Currently writing the three prototypes for a more concrete comparison, but we want to be able to switch to a new one in the future without affecting Run 3 user tasks.

Major data format caveat

- **Tracks vs. vertices ambiguities:** different ways to represent uncertainty
- **1:1 mapping impossible:** we cannot run as they are current code with any new format



Make ambiguous tracks available in both events



Store ambiguous tracks in a separate container

Development areas

Data format

Low deserialization cost

Efficient in-memory store

Optimized decompression

Analysis facilities

Only analyze local data

Fast local storage and network

Allow inter-nodes communication

Workflow handling

Allow for non-linear workflows

Nodes subscribe to data

Use network and shared memory

User interface

Reuse standard interfaces

Declarative paradigm

Optimize common operations

Large scale operations

Interface to central File Catalog

Compose analyses into trains

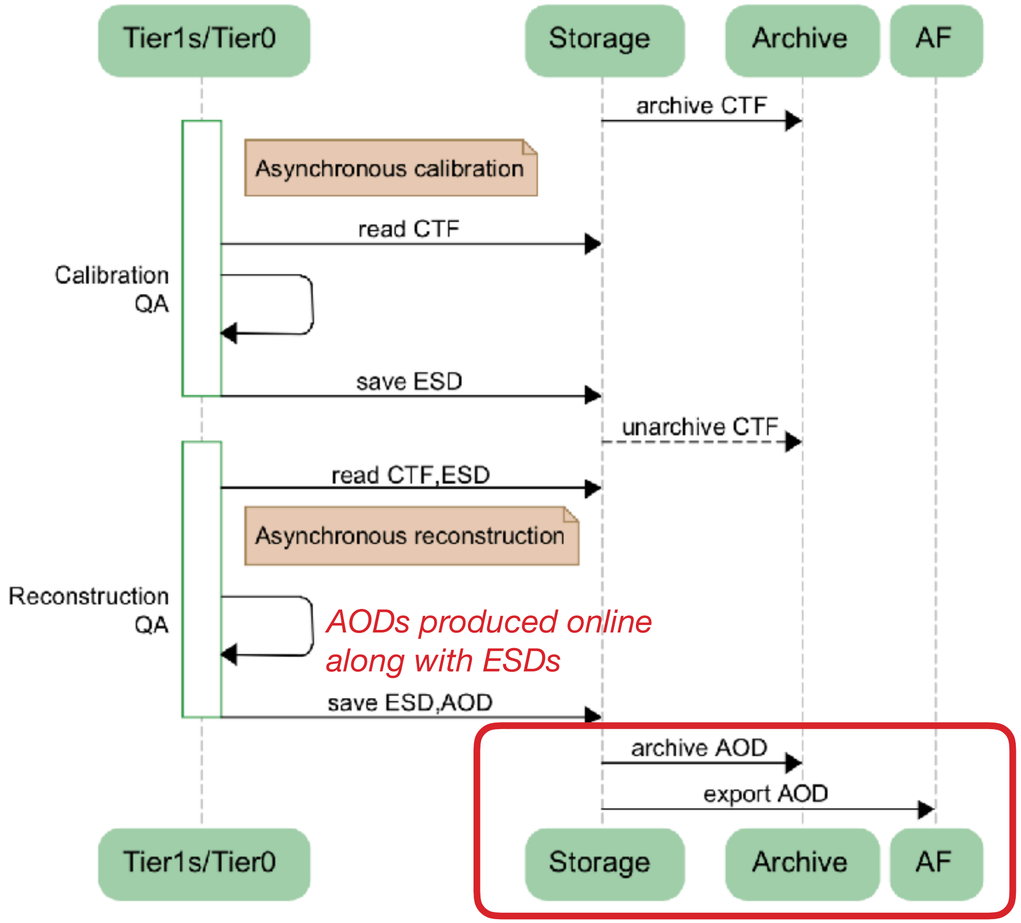
Test and monitor user jobs

Analysis facilities

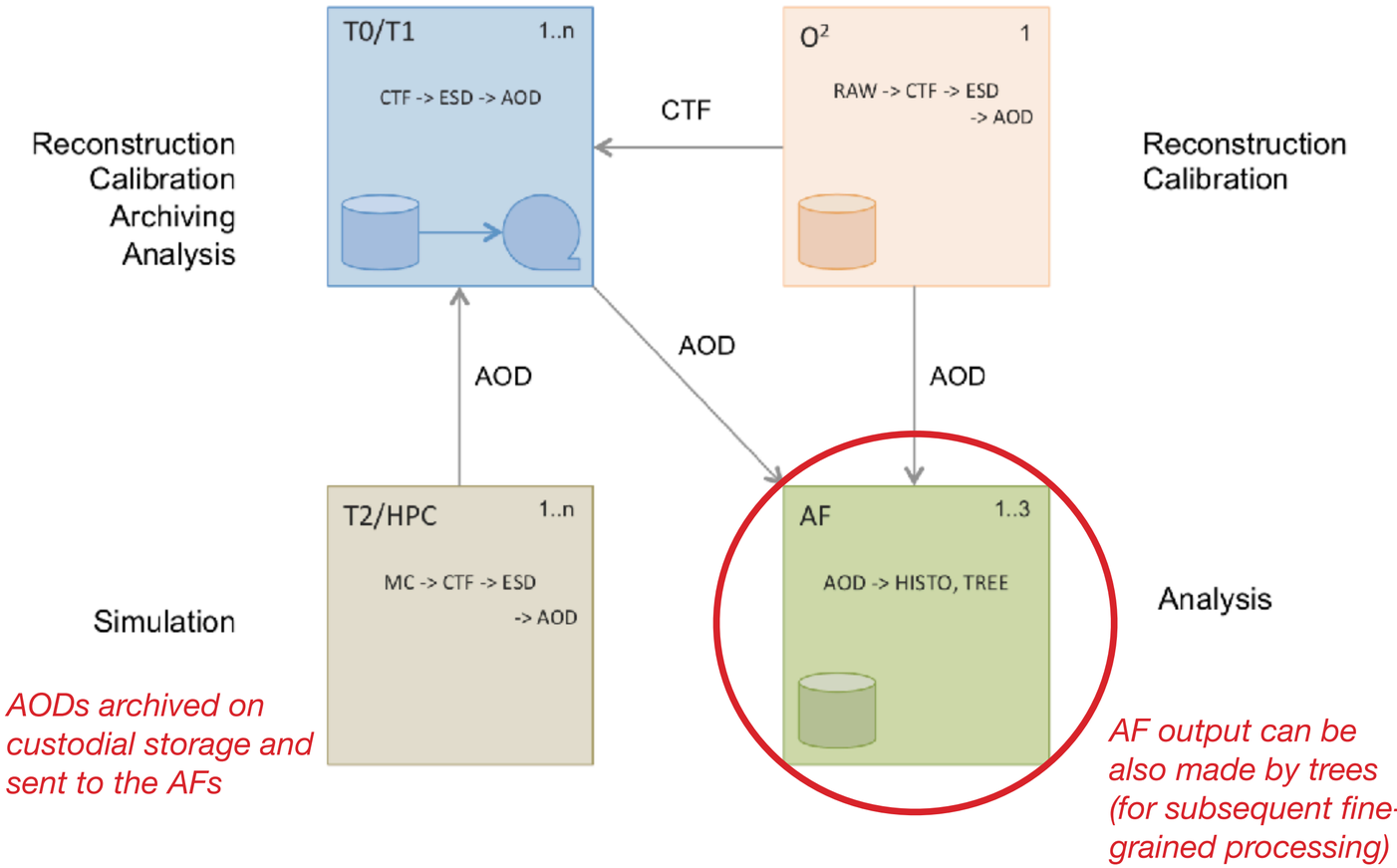
In order to sustain a high-rate of process intercommunications and minimize data reading latency, specially designed analysis facilities are foreseen

- **Intentions:** ALICE will provide for 2/3 large analysis facilities, aggregating ~20k cores each, aiming to maintain the 5 MB/s read throughput. Aggregated throughput: 100 GB/s
- **Local storage:** a large (5 PB) cache managed centrally and accessible via the file catalog too (but local access will bypass it for efficiency)
- **Fast data turnaround:** data constantly replaced with new data based on our convenience. Datasets will have different life spans. Possible to restage data in the future
- **Output results:** output data size negligible. Stored on the AFs, and accessible from everywhere through the central file catalog. Output never purged (differently than AODs)

Data flow up to the analysis facilities



AODs archived on custodial storage and sent to the AFs



AF output can be also made by trees (for subsequent fine-grained processing)

Two or three Analysis Facilities with local storage for organized AOD processing

TDR

Analysis Facility test setup at GSI: benchmarks

ALICE has a test Analysis Facility at GSI (Darmstadt, DE). It fulfills our Run 3 requirements and allows for testing, while currently running Run 2 jobs as a Tier-2

- **Test conditions:** running *current* Run 2 framework on 1500 cores
- **Fast network:** 10 GB/s between nodes
- **Storage:** 15 PB Lustre with 78 OSS, 7 OST, 2 MDS
- **Benchmark:** run operations on 1500 cores. Read results for *each* core:
  - File copy: 1200 MB/s
  - Unzip: 100 MB/s ⇒ must be done in parallel on separate processes!
  - Running the Run 2 framework alone: 20 MB/s ⇒ framework + deserialization cost a lot!
- Further optimization possible via a simpler data format and a novel workflow manager

Development areas

Data format

Low deserialization cost

Efficient in-memory store

Optimized decompression

Analysis facilities

Only analyze local data

Fast local storage and network

Allow inter-nodes communication

Workflow handling

Allow for non-linear workflows

Nodes subscribe to data

Use network and shared memory

User interface

Reuse standard interfaces

Declarative paradigm

Optimize common operations

Large scale operations

Interface to central File Catalog

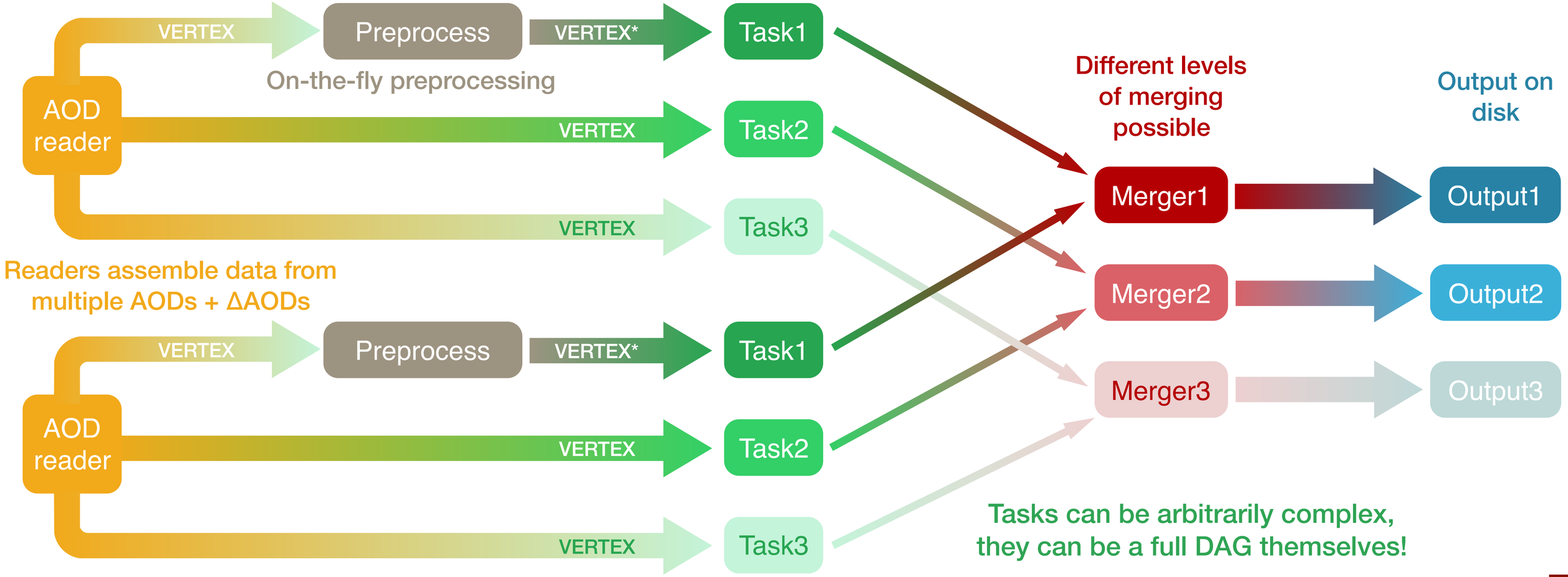
Compose analyses into trains

Test and monitor user jobs

Workflow handling: the Data Processing Layer

File reading is expensive: readers unzip once and dispatch, tasks subscribe to data

ALICE Run3 software is based on the Data Processing Layer: independent processes exchanging data over the network using message queues. Using it for analysis too



Development areas

Data format

Low deserialization cost

Efficient in-memory store

Optimized decompression

Analysis facilities

Only analyze local data

Fast local storage and network

Allow inter-nodes communication

Workflow handling

Allow for non-linear workflows

Nodes subscribe to data

Use network and shared memory

User interface

Reuse standard interfaces

Declarative paradigm

Optimize common operations

Large scale operations

Interface to central File Catalog

Compose analyses into trains

Test and monitor user jobs

User interface: the new ALICE Analysis Task

- **Current analysis:** very simple abstraction, only one degree of freedom: user writes a function for “processing” an event (whatever “processing” means)
- **Declarative approach:** allow user to specify how to filter data and preprocess it first, and then where to store the results, in a compact way
- **Transparent optimization:** by using high-level declarations, optimization heavy lifting is performed by the framework
- **ROOT’s TDataFrame:** nice ROOT development (experimental) allowing to do:

```
TDataFrame d(input).Filter(criteria).Foreach(action)
```

  where low-level optimization and event loop (multithreading, multicore) occur automatically

Currently writing a TDataFrame source allowing to read from Apache Arrow (and evaluating its maintainability). Once again, users don’t deal with the underlying format

Development areas

Data format

Low deserialization cost

Efficient in-memory store

Optimized decompression

Analysis facilities

Only analyze local data

Fast local storage and network

Allow inter-nodes communication

Workflow handling

Allow for non-linear workflows

Nodes subscribe to data

Use network and shared memory

User interface

Reuse standard interfaces

Declarative paradigm

Optimize common operations

Large scale operations

Interface to central File Catalog

Compose analyses into trains

Test and monitor user jobs

Large scale operations

- **ALICE Run 2 analysis is organized:** users write analysis tasks that are assembled as wagons in a train: data is read once and processed by each task serially
- **Even more organization in Run 3:** reading is so expensive that we should do it even less and exploit network message passing of read data
- **Avoid backlogs:** profile analysis tasks to optimize trains composition and avoid slower tasks to fill the reader caches too much and make faster tasks to wait
- **Zero or little re-runs:** current trains can be run several times on the same datasets, but Run 3 data turnaround is shorter. Use trains to filter and run faster iterations on laptops
- **File access:** even if file access is local, a central file catalog is kept. A technique (XRootD plugin) was developed to bypass catalog access for local access wherever possible
- **Trains web frontend** is currently very popular for composing and running trains easily, we will need to adapt it to the new framework

What's to be done on the short term

Short term deadline is July (CHEP): we aim to have a sample analysis running as a demonstrator by then. Converging on the data format is the most important first step

- **Several small components ready:** analysis facility test setup, TDataFrame, three different data format libraries, a data processing layer handling the workflow
- **Implement the base data format** using the given libraries for comparing their speed. Implement macros to convert sample data to the new formats automatically (continuous integration fashion)
- **Develop the appropriate base data processing layer tasks:** a file reader/unzipper and the actual task, based on the TDataFrame interface
- **Deploy a sample workflow topology** on the test Analysis Facility
- **Develop the appropriate TDataFrame sources** once we have benchmarked them

Thanks!

National distributed disk cache for CMS@LHC: Status and Progress

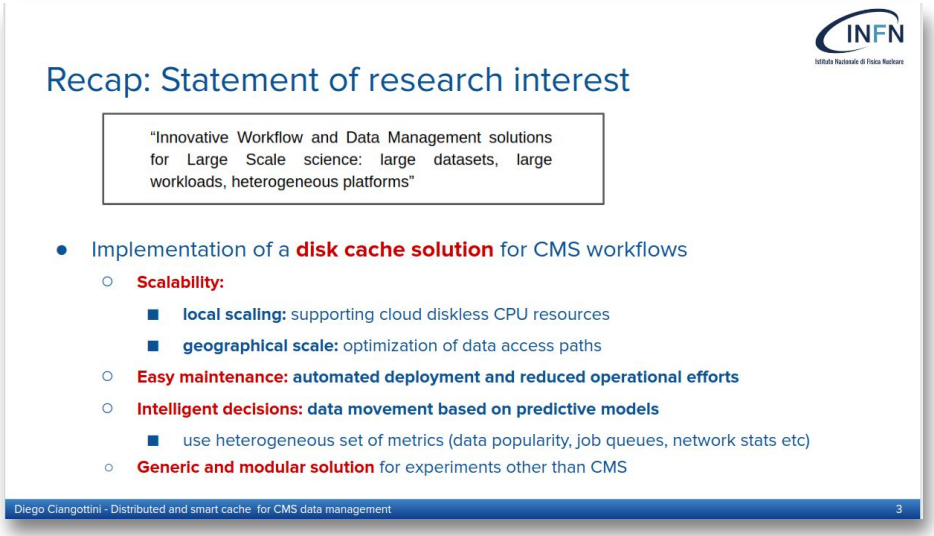
Diego Ciangottini[a]

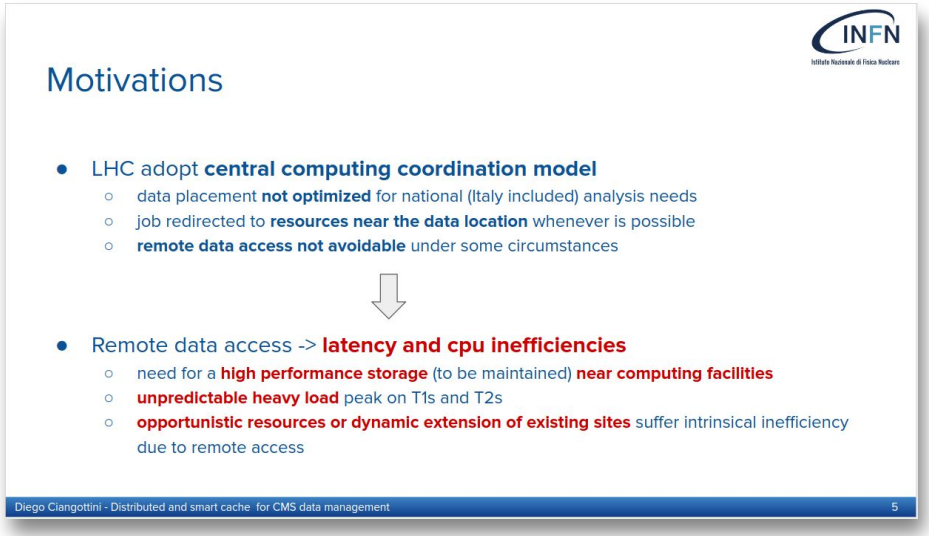
a) INFN, Perugia

Outline

- Activity motivations remind
- XRootD cache architecture
- Current status
- Next steps
  - Deployment of a first National distributed testbed
- Long term view
  - Data-lake integration

Motivations





CSN1 meeting 21-23 Feb

https://agenda.infn.it/getFile.py/access?contribId=27&resId=0&materialId=slides&confId=14896

A cache system to support several scenarios

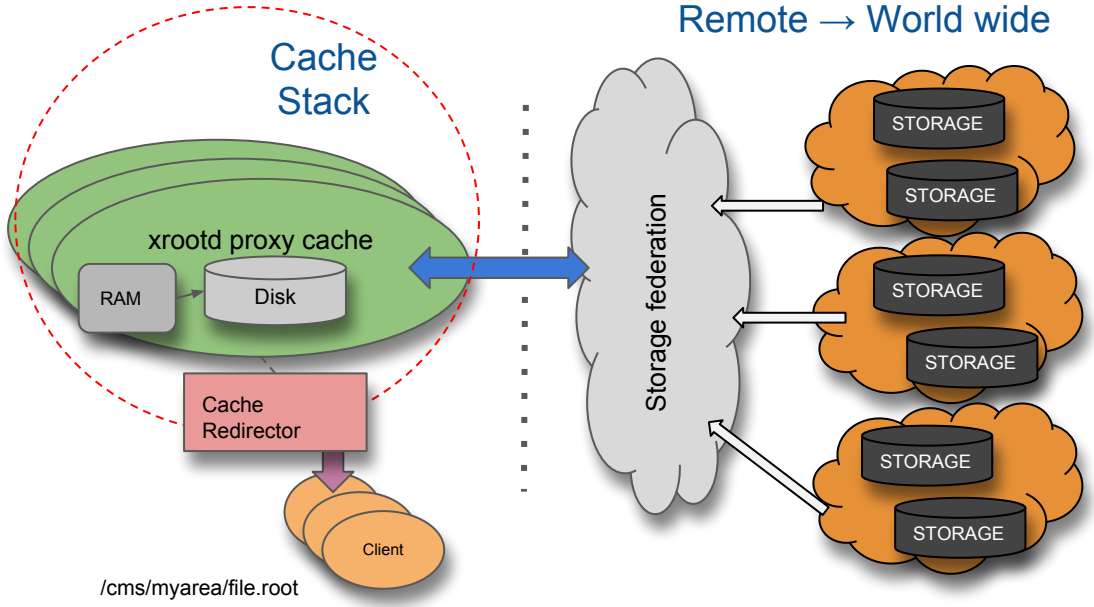
- **Leverage national networking to reduce total maintained storage resources**
- **“Data-lake” approach:**
  - **Interposing cache** on top of **a central custodial site**
- **Opportunistic Computing:** to bring **not pledged resources** in the computational model
  - From the experiment point of view: to integrate cloud based resources with **zero effort**
- **Dynamic Site Extension:**
  - **Peak of usage** or more in general buying **external cloud resources:**
    - see activities like: Aruba, Microsoft Azure, HNSci project etc

NOTE: The presented activity lives within the CMS Data Management project

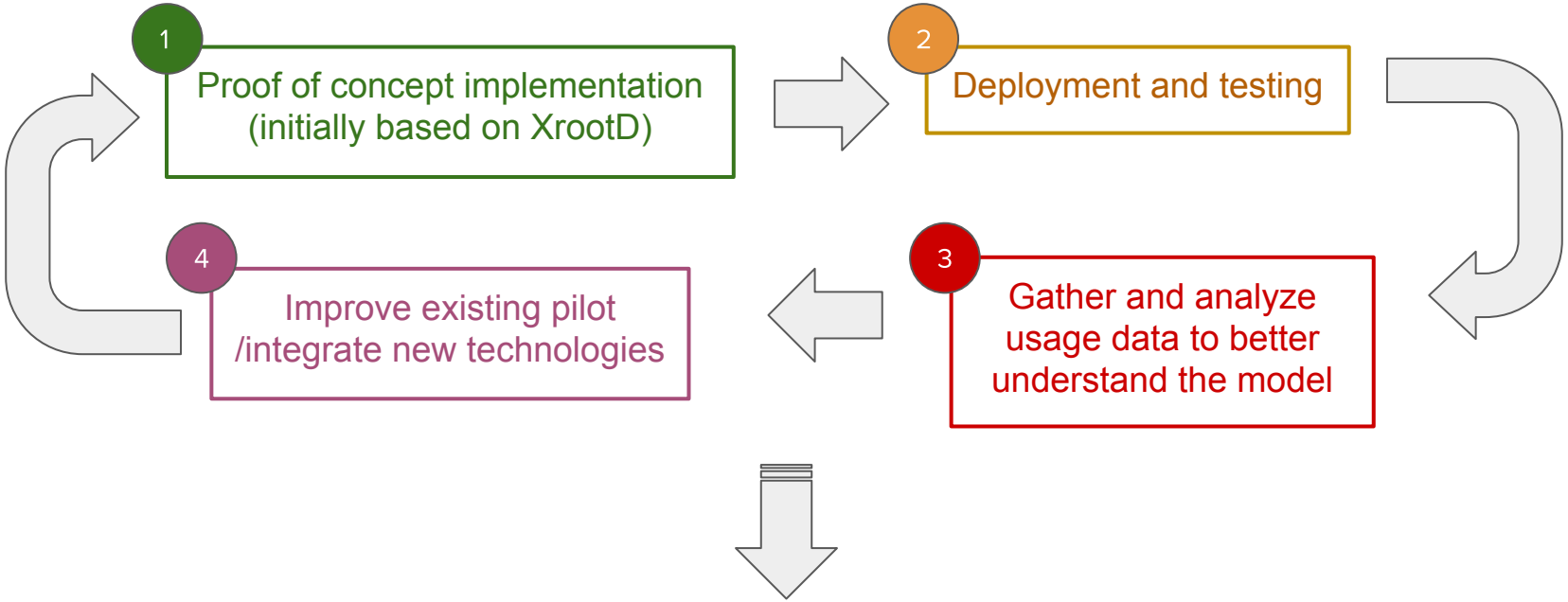
XCache implementation

XRootd based implementation of a disk proxy-cache tool

- XRootD infrastructure **spans all of the Tier-1 and Tier-2 sites in EU and US CMS**
  - **well known protocol at computing sites**
- **Multiple storage backend** support and optimization
- **Easy integration on current LHC computing model**

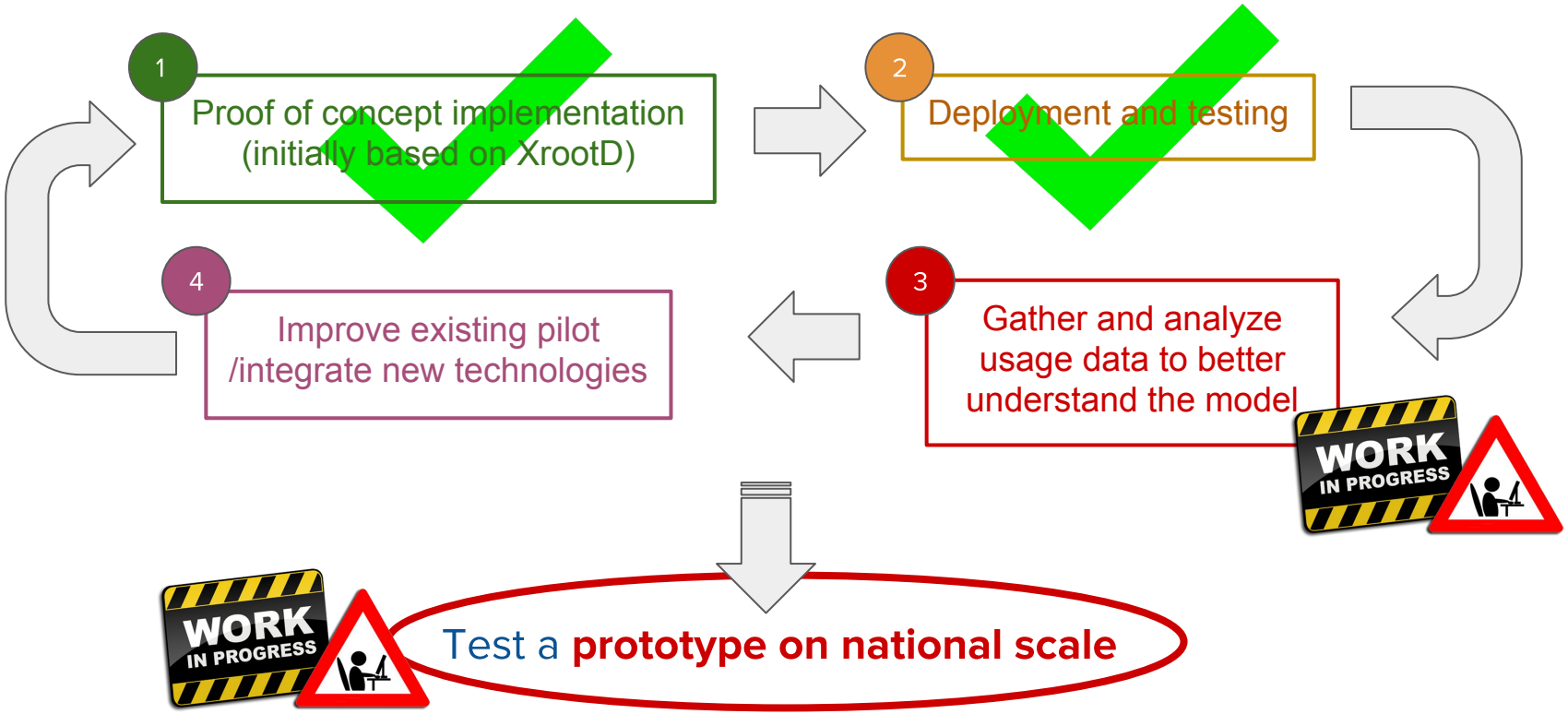


Work strategy



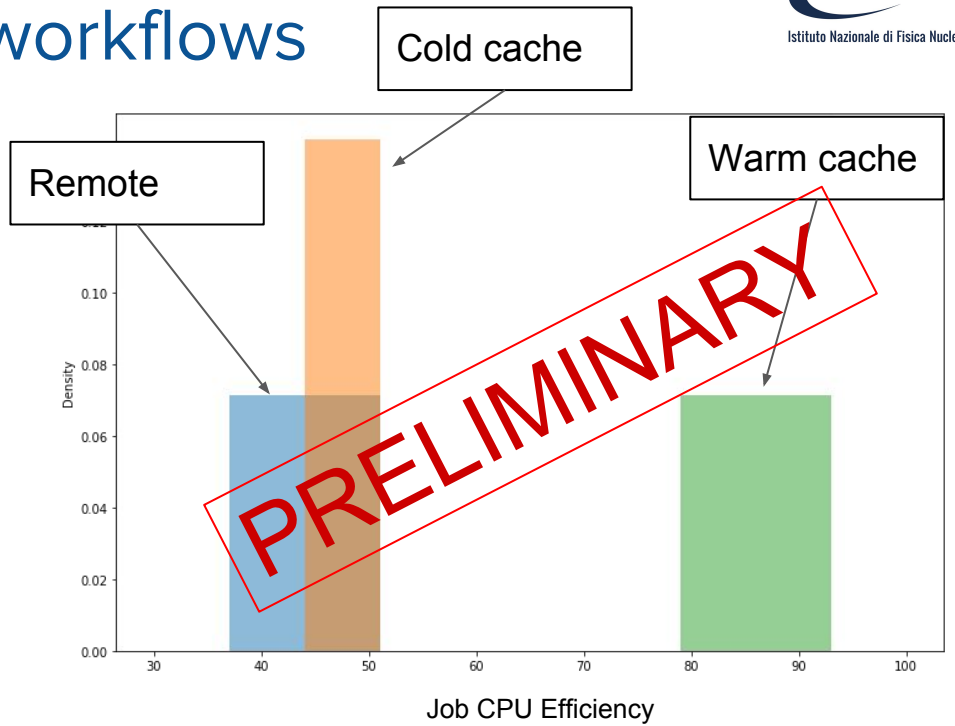
Test a **prototype on national scale**

Work strategy: status update



Testing cache with CMS workflows

- A first **proof of concept** has been developed and **first tests with CMS analysis workflows** are ongoing.
  - **DODAS environment** to provide a recipe for an **automatic XrootD cache deployment on heterogeneous cloud resources**
  - utilize “any cloud provider” with almost zero requirements and a **simple text configuration file.**
    - **Open Telekom Cloud (OS based)** resources used for preliminary results



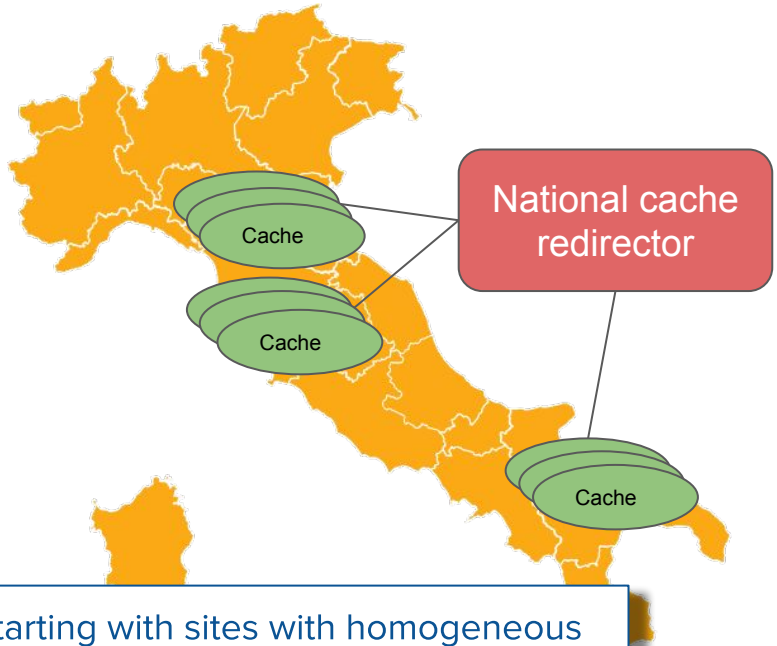
Everything behaves as expected.

Remember that the amount of efficiency gain is heavily **workflow dependent**

National testbed deployment

- **Objective: to deploy a national level cache**
  - **geographically distributed** cache servers
  - **heterogeneous resources and providers**
  - Leverage **national networking to optimize the total maintained storage** resources
- **Collection of important data for evaluating the benefits on a realistic scenario**

Already contacted CNAF, Pisa and Bari to support **the deployment of a national testbed for Xcache federation → Agreed!!**



Starting with sites with homogeneous resources (gpfs-storm).
Then extending to other sites (e.g. Legnaro) on a second step.

National testbed deployment activity

- Reproduce on national scale the **same architecture and tests as on cloud resources**
  - cluster of cache servers federated under a dedicated redirector
- Write a **technical proposal for activity coordination**
- **Functional tests** with **no dedicated high IO hardware** (e.g. ssd etc)
  - in general no additional costs, using hardware that is already in place

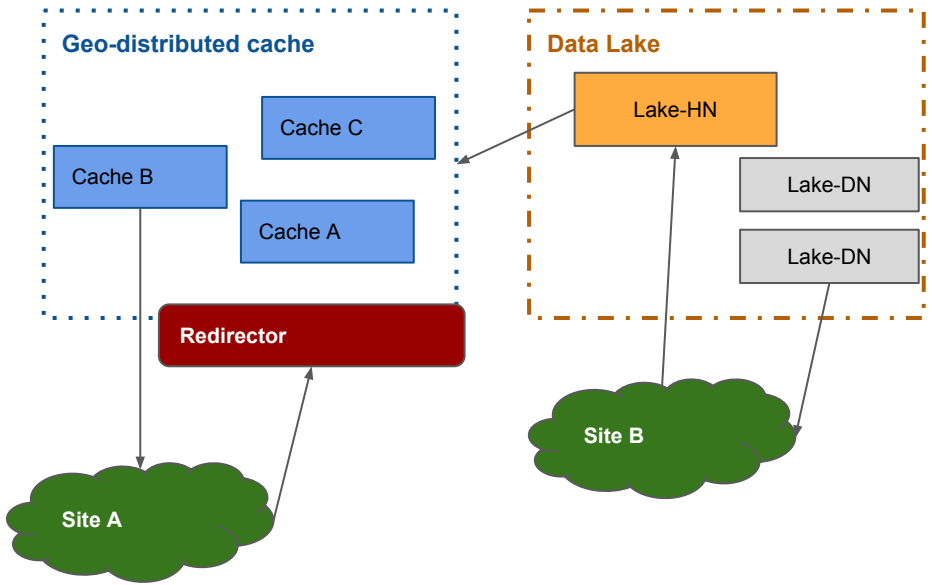
XCache and Data-Lake

The activity on Italian testbed can be **a first benchmark for future solutions** proposed in an LHC data-lake scenario.

Synergies:

- outlook on **projects with similar motivations and objectives** (**eXtremeDataCloud** and others if any will come up)
- with **CERN investigation of XCache application for internal EOS cache mechanism.**

- Lake-HN : central service of the data lake (namespace, metadata etc)
- Lake-DN : storing data and under the management of Lake-C



Summary

- **INFN XCache for CMS@LHC is proceeding as for roadmap**
  - first phase ready ➔ automation and local cluster test
- The technology used is **based on XRootD**
  - **multi-backend storages** ➔ generic application
  - every **system XRootD compliant can use XCache**
  - possible to be extended **beyond CMS boundaries**
- **Existing synergy with pure HTTP approches** (Sonja talk today)

My two major milestones for 2018:

- **Tests with national testbed**
- **Dynamic site extension:**
  - e.g. **overflow:** CMS payloads assigned to CNAF pledged resource ➔ redirected seamlessly to cloud resources deployed with DODAS

Backup

Motivations and features: recall

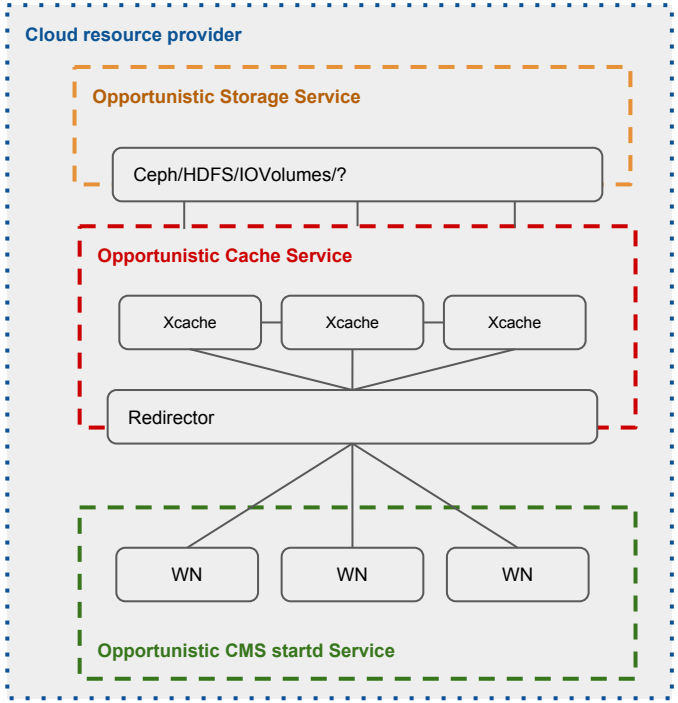
- **Reduce latencies / Improve efficiency** on remote data access
  - enhance CPU efficiency and job success rate
- **Reduction of traffic load**
  - mitigate the load on custodial storages
- Optimization of data access for **opportunistic resources**
  - e.g. sites extension, public cloud etc..

Important features:

- **Scalability**
- **Easy maintenance**
- **Intelligent decisions**
- **Modular solution**

Architecture outlook

- **Modularity**
  - factorized applications
  - cache on top of existing storages
  - seamless scaling
- **Packaging**
  - docker images
  - health-checks for self-healing implementation
- **Plug-in**
  - cache algorithms
  - clustering data distribution



XCache in CMS

- **Reduce latencies / Improve efficiency** on remote data access
  - enhance CPU efficiency and job success rate
- **Reduction of traffic load**
  - mitigate the load on custodial storages
- Optimization of data access from **opportunistic resources**
  - e.g. sites extension, public cloud etc..

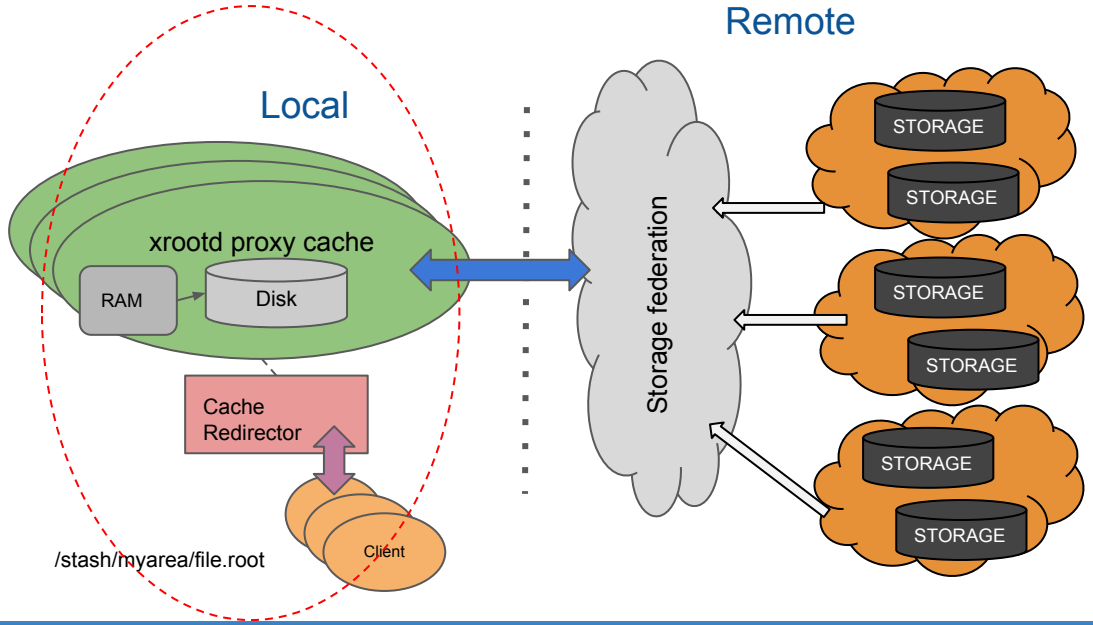
Main features

- **Scalability:**
  - **local scaling:** supporting cloud diskless CPU resources
  - **geographical scale:** optimization of data access paths
- **Easy maintenance:**
  - automated deployment and reduced operational efforts
  - self-healing
- **Intelligent decisions:**
  - data movement based on predictive models
- **Modular solution:**
  - easy to extend and pluggable

Local site scenario

e.g. opportunistic sites and remote site extension

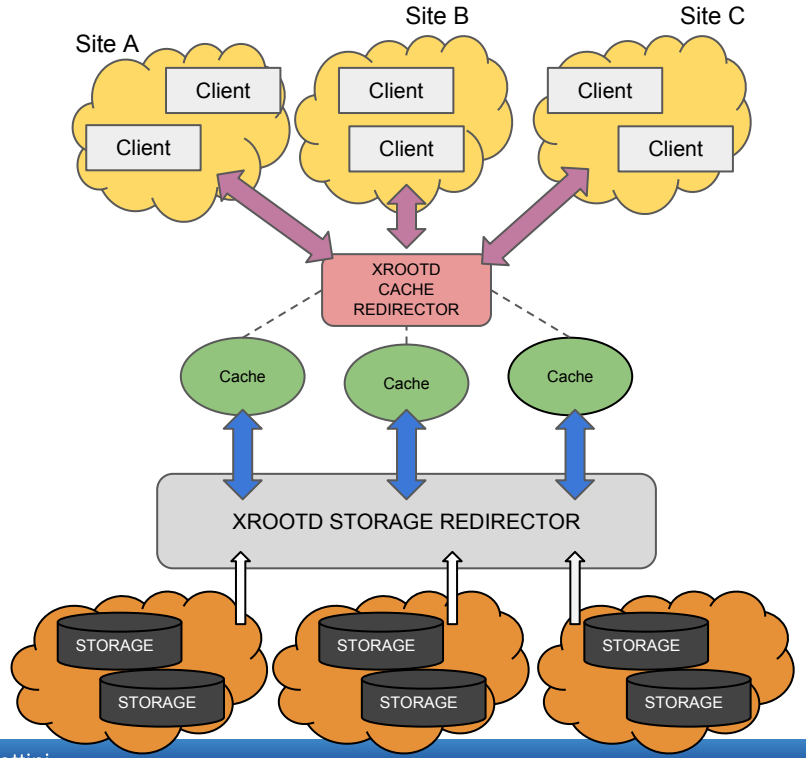
- Create a **cache layer near cpu resources**
- Bring it up **on demand**
- **Scale horizontally**
- **Federate caches in a content-aware manner**
  - redirect client to the cache that currently have file on disk



Distributed scenario

Geographically distributed cache

- The very same technology used on local scenario can be **geo-distributed**
- Use **ephemeral storages to enhance jobs efficiency**
- Leverage high speed links to **reduce the total amount of allocated space**



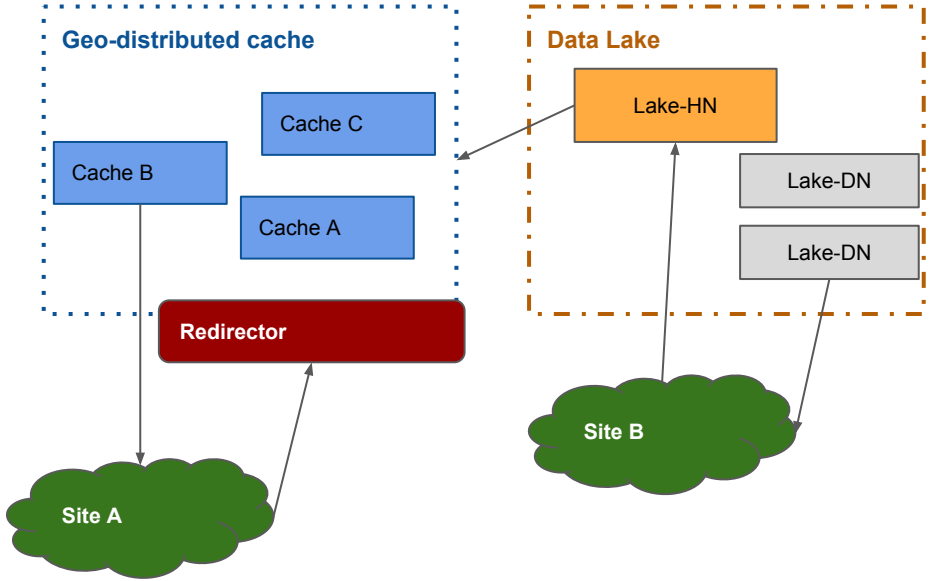
Side note on future development

Data lakes model

- **Geo-distributed scenario** is also **part of a “data lake” model**

- **creating a multi-site cache layer**

- Lake-HN : central service of the data lake (namespace, metadata etc)
- Lake-DN : storing data and under the management of Lake-C



Current state of the work

- Implemented **local site scenario:**
  - Preliminary functional tests
  - **Local scale scenario test on cloud resources**
    - deployed on private and public cloud
    - setup automation
    - CMS workflow test

Configuration and integration overview

Example

```
sudo docker run -v $PWD/config:/etc/xrootd cloudpg/xrootd-proxy --config /etc/xrootd/xrd_test.conf
```

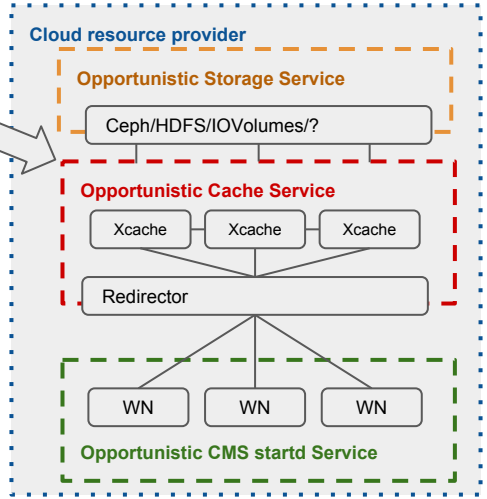
- **CMS Xcache Docker container** has been setup to allow an easy deployment
  - passing a **complete xcache config file**
  - or setting **caching parameter as arguments/env**
  - healthcheck call implemented

A Docker Compose configuration file is available to **orchestrate the deployment of a local test instance.** The stack contains a test remote server, a cache instance and cache redirector (preliminary docs here)

- then a variety of recipe for **orchestration tools have been evaluated**:
  - **docker swarm, k8s and marathon services** (redirector+caches)
  - config and scale services with compose-like recipes

swarm orchestration kubernetes MARATHON

- **Open issue: authentication**
  - the cache server **authenticate with remote storage through its own credentials**
  - **no user cred forwarding**
  - **no token authentication yet**



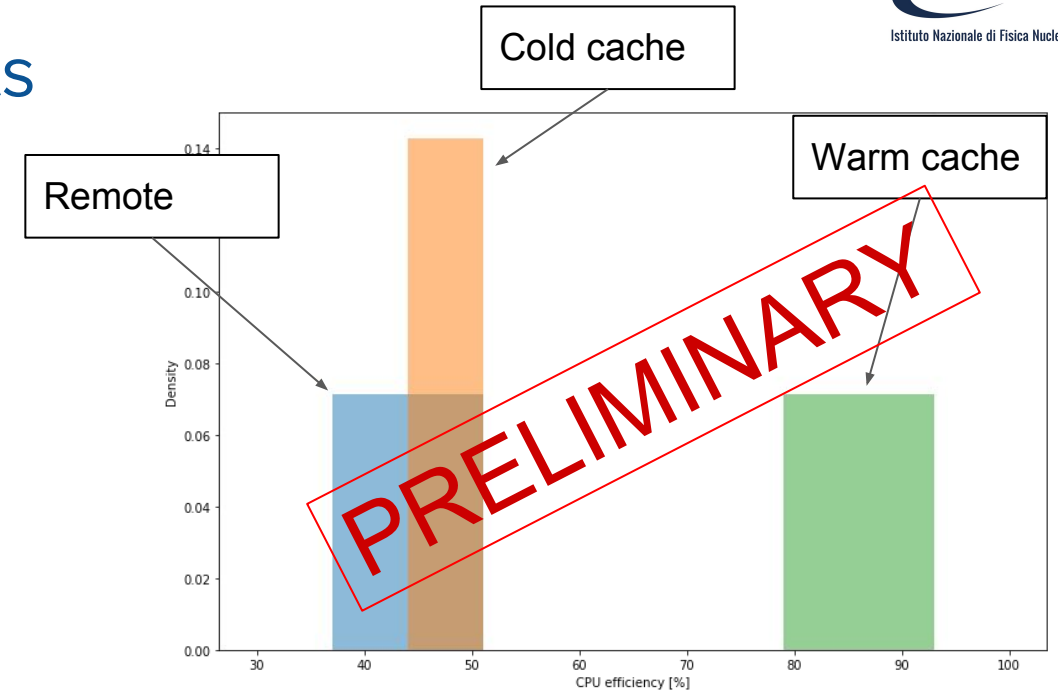
Tests on local site scenario

- **Tests with CMS analysis workflows**
  - **DODAS** service have been used
    - same configuration for setup on different cloud providers
    - automated deployment

- **Measurement and comparison of CPU efficiency with:**
  - remote data access
  - cold cache (missing file)
  - warm cache (file found)

First benchmark results

- **1 cache server**
  - Open Telecom Cloud
  - VM 16 cores and 32GB RAM
  - 500GB HighIO flavor
  - Cache config:
    - prefetch: 0
    - block size: 512k
    - origin: xrootd.ba.infn.it
- **4 WNs** over the same internal network
- **IO test with CMS analysis workflow**
  - 4 jobs
  - reading vertex associated tracks information
  - **ad-hoc setup to enhance the network latency effect**

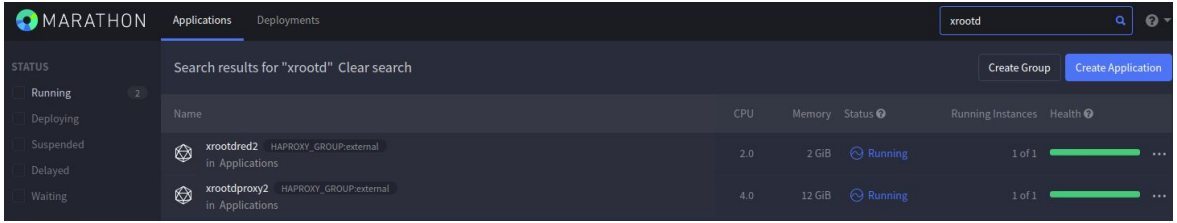


Everything behaves as expected.

Remember that the amount of efficiency gain is heavily **workflow dependent**

Cache management and monitoring

Scaling up and down cache servers dynamically

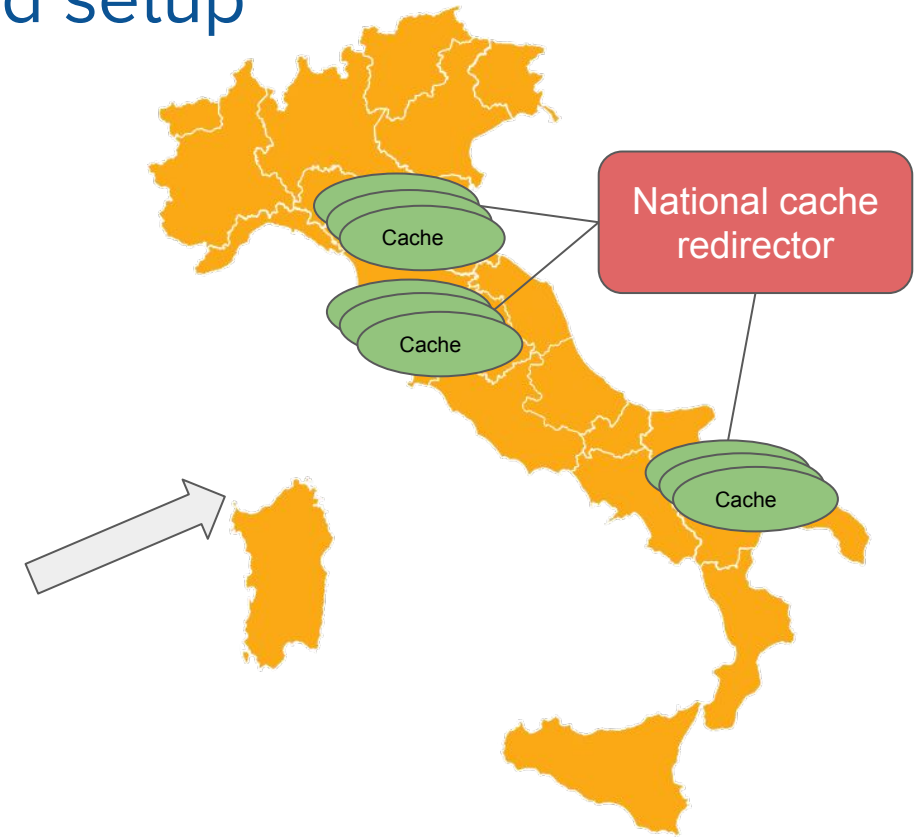


The deployment provides an **integrated monitoring stack** based on Elasticsearch Beats and Grafana Dashboards

Planned activity: distributed setup

NEXT STEPS

- get **quantitative evaluation** of cache performance on **local scenario**
  - different WFs, configuration, backend etc
- **distributed scenario prototype (national level)**
  - **geographically distributed** cache servers
  - **heterogeneous resources and providers**

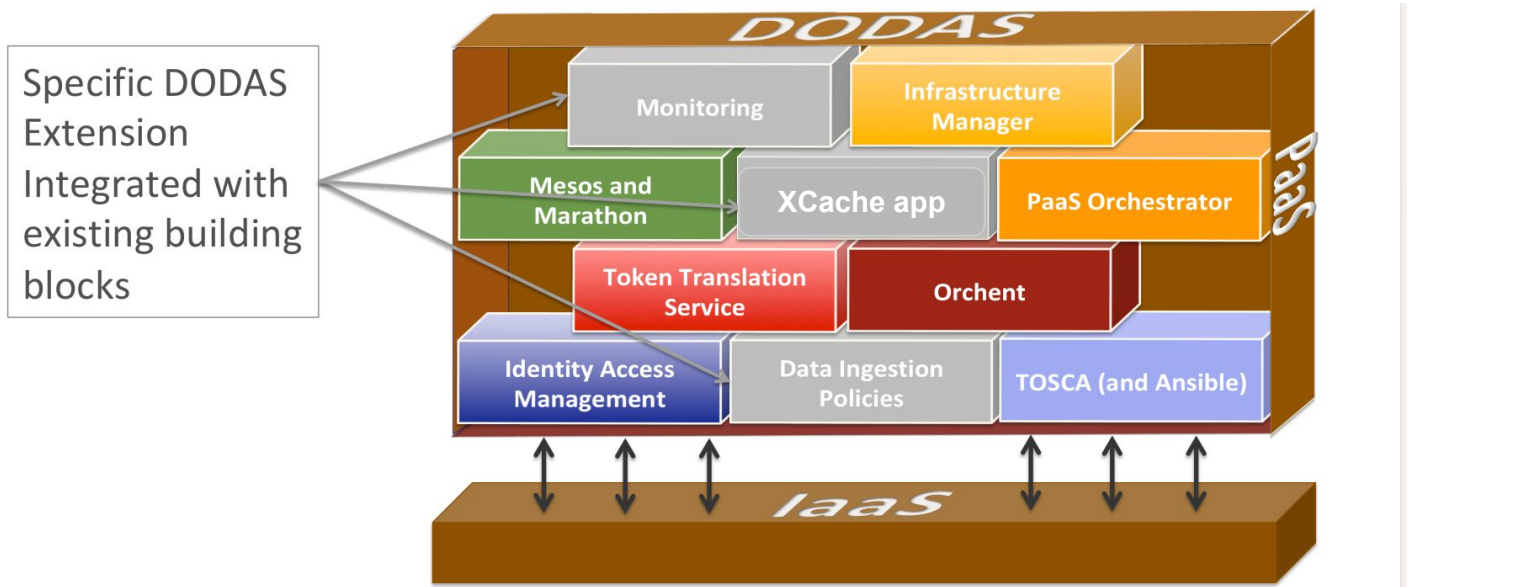


Summary

- Deployed a first prototype of **XCache instance on private and public cloud provider**
- Started to **measure performances on a benchmark workflow**
- **Recipes for automatic deployment** of XCache on cloud resources
- Do we find sinergies with similar activities in CMS ?

IaaS overlay: DODAS integration

- Service for generating over cloud resources an on-demand, container based application deployment.



Tests on local scale scenario

What's needed?

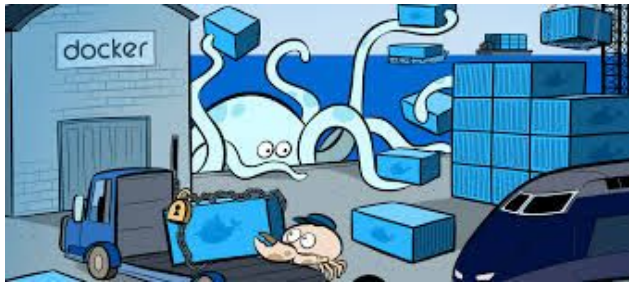
- **XRootD cache packaging**
  - **Docker container** with different configuration
- **Containers orchestration**
  - different solution available: **docker swarm, k8s and marathon pods** (redir+caches)
- **Working at PaaS level**
  - tests with **DODAS integration**

CMS XCache packaging

A **CMS Xcache Docker container** has been setup to allow an easy deployment

Example

```
sudo docker run -v $PWD/config:/etc/xrootd cloudpg/xrootd-proxy --config /etc/xrootd/xrd_test.conf
```



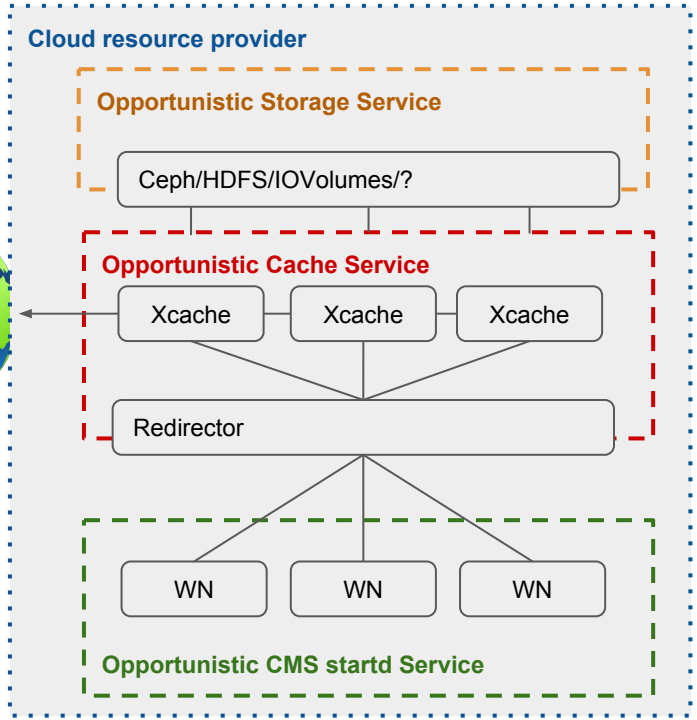
A Docker Compose configuration file is available to **orchestrate the deployment of a local test instance.**

The stack contains a test remote server, a cache instance and cache redirector (preliminary docs here)

Orchestration

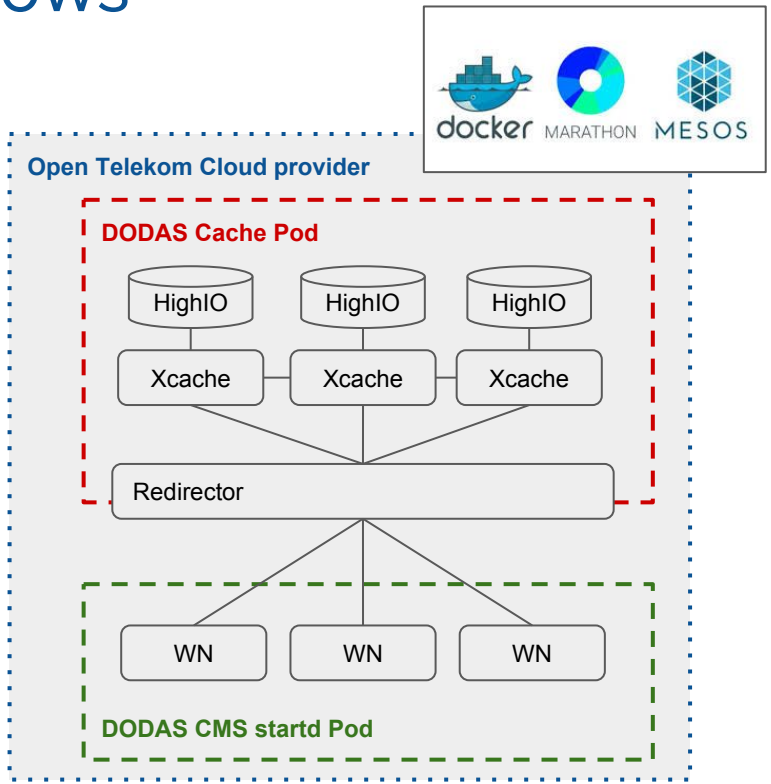
A variety of orchestration tool can be used for the deployment:

- **Docker swarm**
  - deploy and scale services on multiple nodes using docker-compose recipes
- **Kubernetes or Mesos+Marathon**
  - deploy and scale cache services containers as pods with a compose-like recipes



Testing cache with CMS workflows

- A first **proof of concept** has been developed and **first tests with CMS analysis workflows** are ongoing.
  - **DODAS environment** to provide a recipe for an **automatic XrootD cache deployment on cloud resources**
  - utilize “any cloud provider” with almost zero requirements and a **simple text configuration file.**
    - **Open Telekom Cloud (OS based)** resources used for the following results



Cache in CMS: XRootD based cache

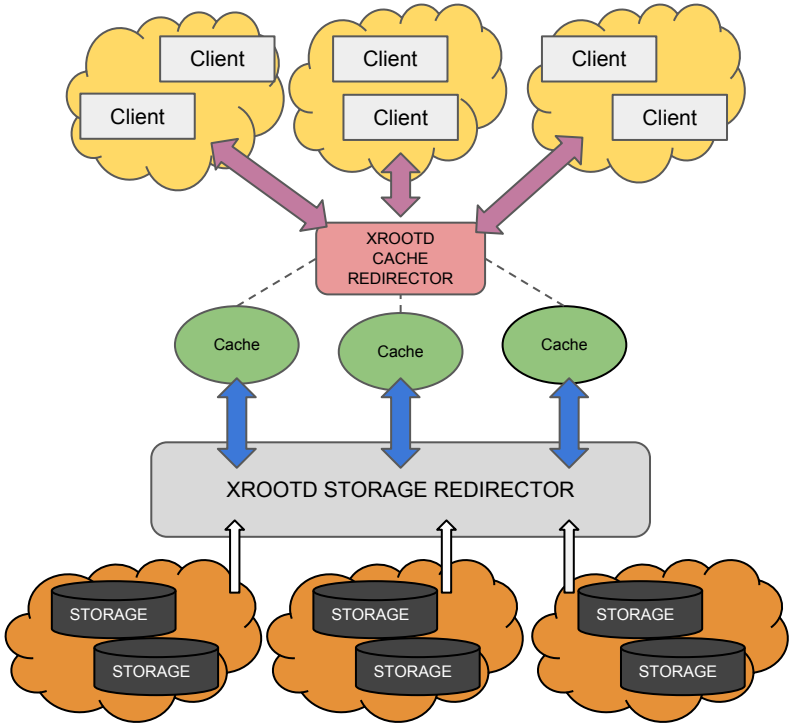
- CMS model can integrate **XRootD caches (XCache) seamlessly**
  - XRootD is widely supported at T1s and T2s
  - XCache is modular and pluggable
  - cluster many caches with a cache redirector
  - already under evaluation by various activities within WLCG

Cache keywords

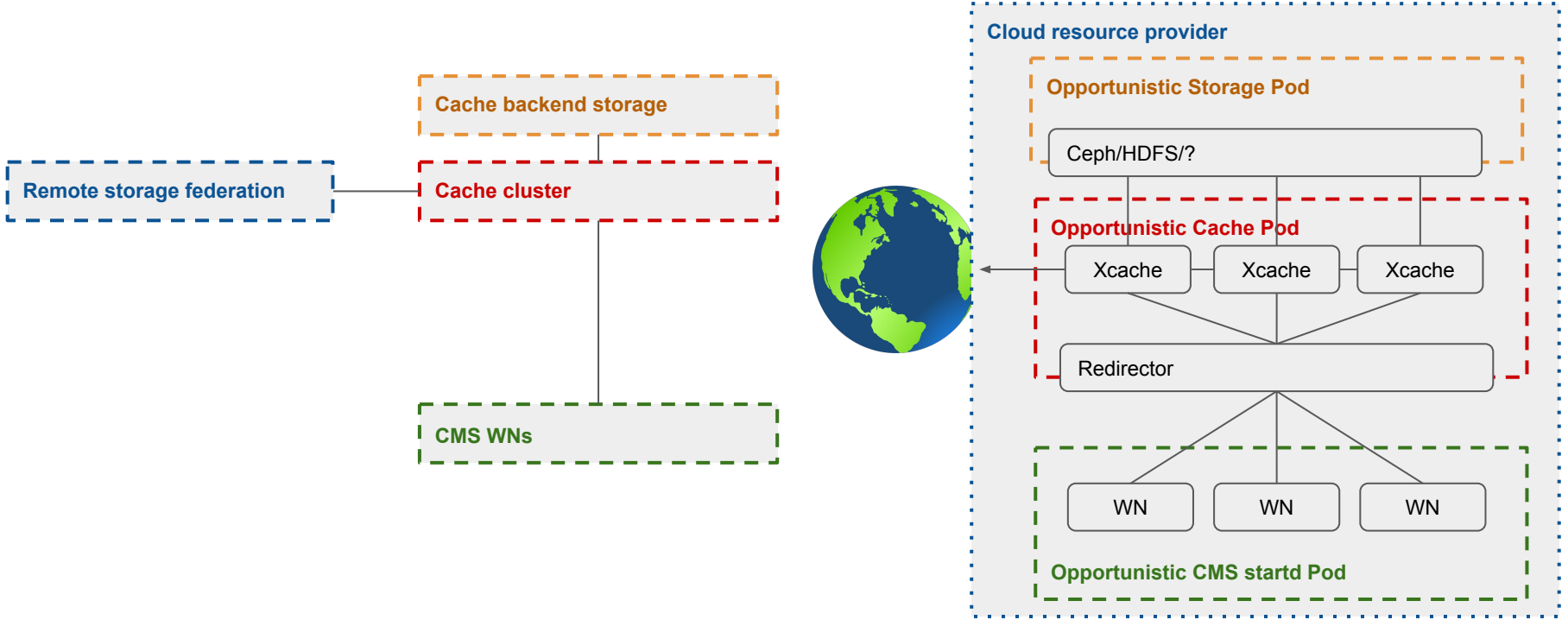
- **Cache metadata**
  - stores details about already downloaded blocks and all local accesses.
- **Prefetching**
  - cache can issue advance read requests to reduce read latency
- **Decision plugin**
  - allows users to configure which parts of namespace are to be cached.
- **Cache purging**
  - e.g. high/low water mark algorithm to start/stop purging. Plugins for smarter algos should be possible

Clustering with xrootd cache redirector

- Through the XrootD redirection is possible to **federate caches in a content-aware manner**
  - redirect client to the cache that actually have file on disk
- **Loadbalancing:** If no cache has the requested file, **a round robin selection of cache server is used** (*configurable*)
- **Overloading:** If a file present on **one cache is requested by many clients**, it can be **allowed to be duplicated on other servers**



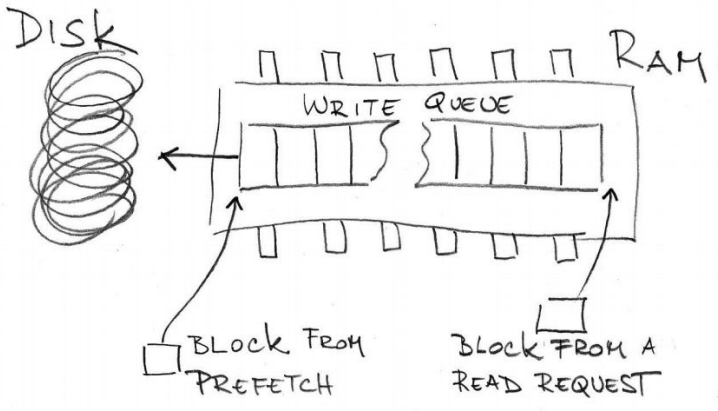
Vision on caching for opportunistic resources



XrootD cache mechanics: write queue

- **Prefetched buffers** are put to the **beginning of the write queue**
  - assume they will be needed at a later time so **RAM should be vacated as soon as possible.**
- **Buffers obtained to serve outstanding read requests** are put to the **end of the write queue**
  - assume they will be needed to serve future read requests so they should be **kept in RAM as long as possible.**

From A. and M. Matevz talk ICEPP2016



XRootd, disk-based, caching proxy for optimization of data access, data placement and data replication. A. T. Bauerdick, L & Bloom, K & Bockelman, B & Bradley, Dan & Dasu, S & Dost, Jeffrey & Sfiligoi, I & Tadel, A & Tadel, Matevz & Wuerthwein, Frank & Yagil, A. (2014). *Journal of Physics: Conference Series. 513. 10.1088/1742-6596/513/4/042044.*

XrootD cache mechanics: overview

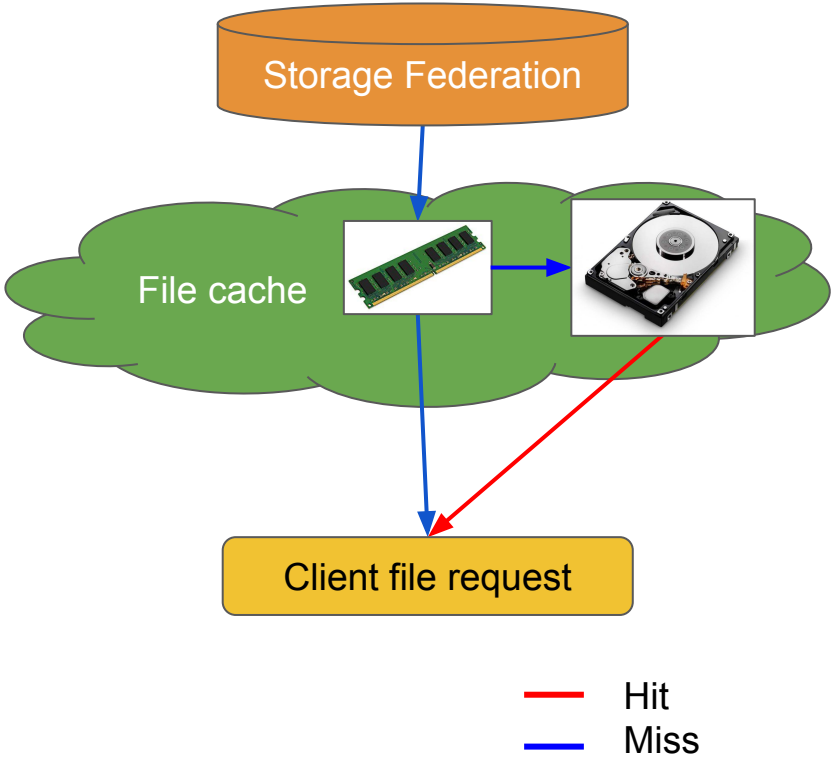
Open File

1. *Cold cache:* remote open through storage Federation
2. *Warm cache:* opens file on local disk

Note: remote open is only initiated if/when a requested block is not available in the cache.

Read File

1. If in RAM/disk→serve from RAM/disk
2. Otherwise request data from remote and
   a. **serve it to the client**
   b. **write it to disk via write queue** (this way data remains in RAM until written to disk)



Dynamic On Demand Analysis Service

Dynamic On Demand Analysis Service (DODAS) is a solution developed in the context of INDIGO-DataCloud project.

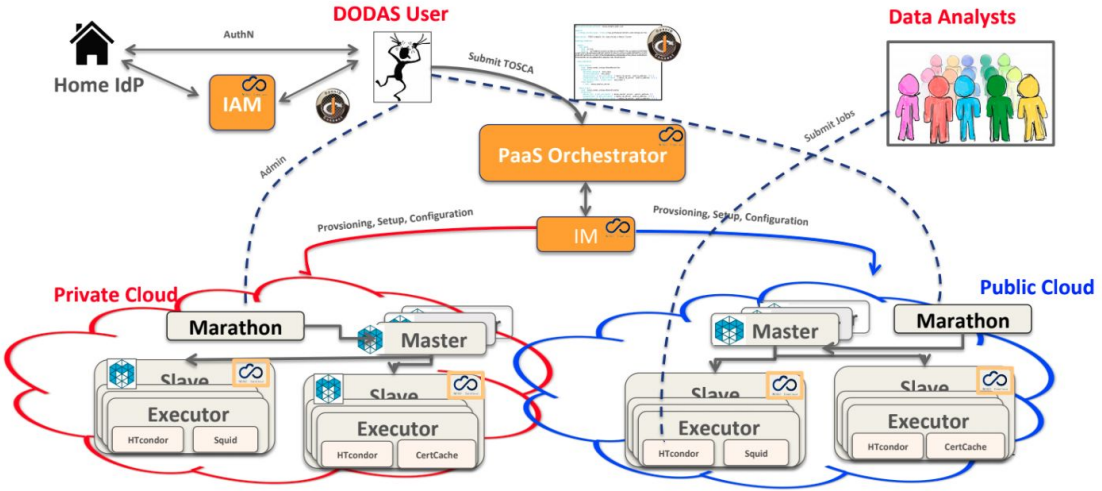
It is a service for generating over cloud resources an on-demand, container based application deployment.

That includes solutions that spans from a standalone HTCondor batch system to a Big Data processing cluster

To support user tailored computing environments

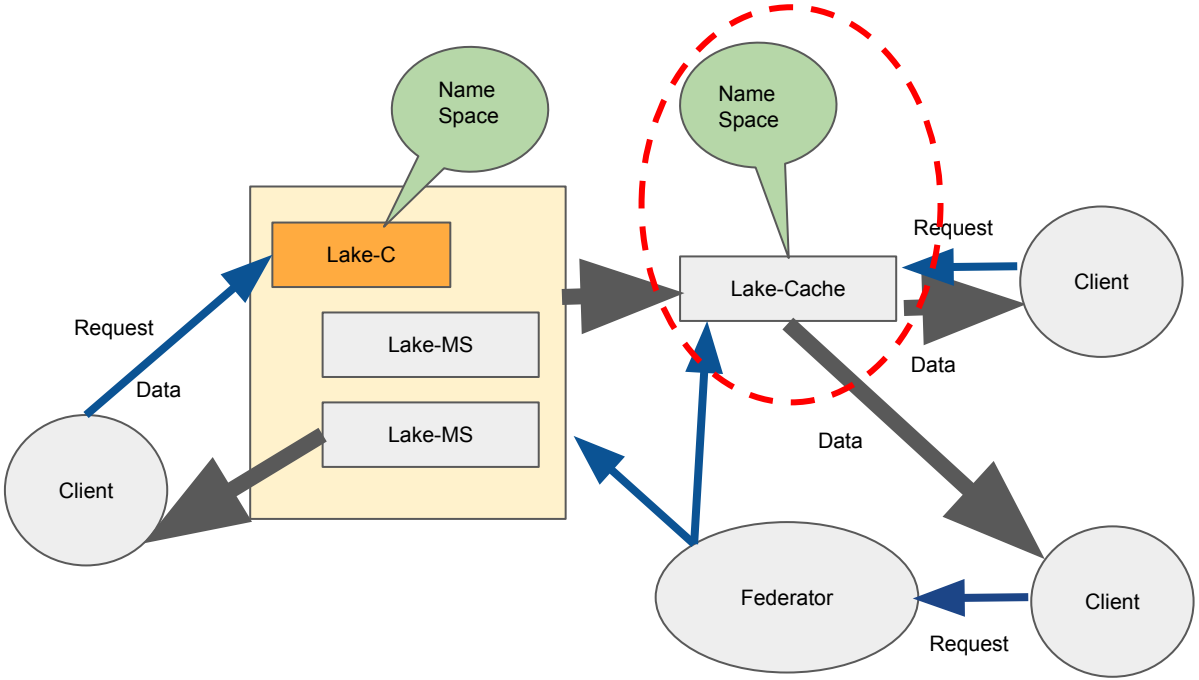
To automate configuration and deployment of custom services and/or dependencies

To define input parameters and customize the workflow execution



XDC scenario

- Lake-C : The central service of the data lake, holding namespace, metadata and making scheduling decisions. The “head node”.
- Lake-MS : A service storing data and under the management of Lake-C. The “disk node”.



Application of Machine Learning strategies for Physics Analysis in ATLAS

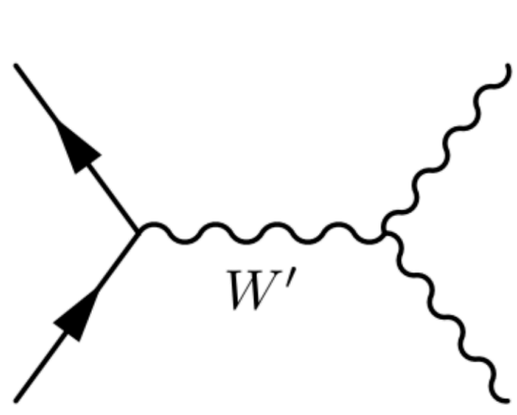
Dinos Bachas
INFN Lecce

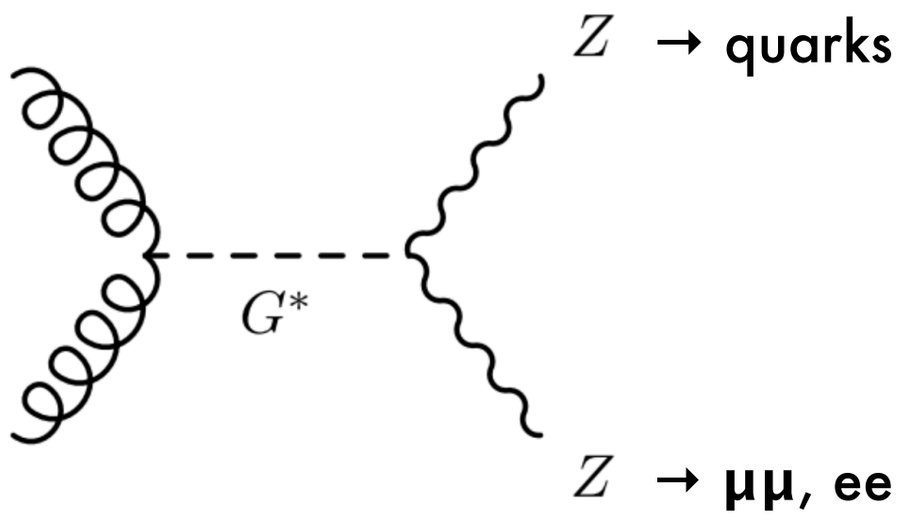
INFN and The Future of Scientific Computing - Episode I: The HPC Opportunity

04 May 2018
Accademia delle Scienze, Turin

Work overview - Area of involvement

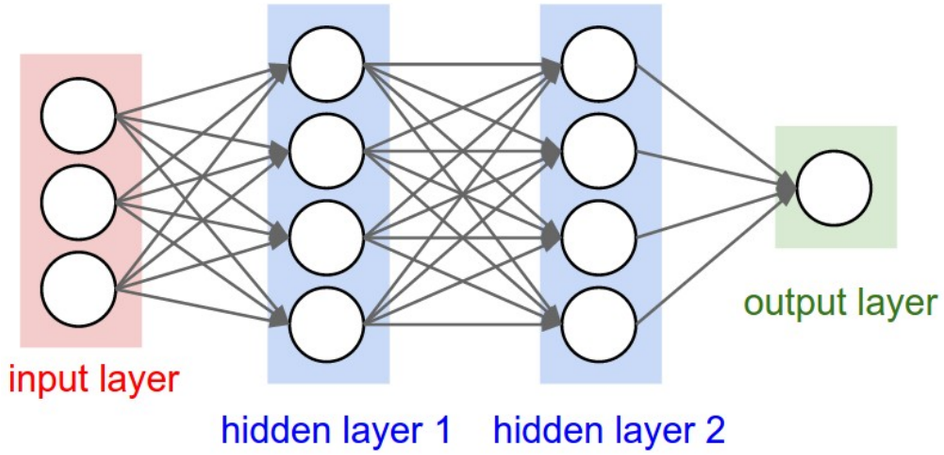
- Working in the context of 'INFN Fellow in Scientific Computing' since mid October 2017 at Lecce station.
  - Collaborators: Stefania Spagnolo, Gabriele Chiodini
- Area of interest: application and development of Machine Learning (ML) strategies to physics analysis at the LHC and ATLAS
- Exotics searches for heavy resonances decaying to dibosons with semi-leptonic final states
  - ZV → ℓℓqq (V = W,Z and ℓ=e,μ) with 3 main classes of signal models: spin-0 (Higgs), spin-1 (W') and spin-2 (G*)
    - Both **boosted** and **resolved** topologies are used to maximize the sensitivity in the intermediate and high mass regions.
- The current signal selection efficiency for the llqq analysis is around 0.4 for the merged selection (at 1TeV mass)
  - Can we do better than that at the same background rejection?
- Try to use ML and Deep Neural Networks(DNN) to improve sensitivity achieved with cut-based analysis. Signal/Background classification problem
- Develop parameterized DNNs to tackle problem of interpolating the DNN results to untrained mass points





Deep Learning for S/B selection optimization in exotics searches

- Build DNNs using open source frameworks, ML libraries, data handling software
  - Keras, Theano, Tensorflow, Pandas, Scikit-learn, Numpy, SciPy
- Train DNNs with variable sets of inputs and architectures
  - Basic kinematic inputs and/or high level features (e.g jet substructure)
  - Investigate combinations of number of neurons, hidden layers, activation functions, learning rate…
- Estimate performance metrics for the DNN for each combination
  - ROC curves, Area Under Curve(AUC), model accuracy…
  - Deal with overfitting using recommended techniques (e.g dropout, K-Fold cross-validation)
- Comparison with cut-based analysis at specific signal efficiency/ background rejection working points
- **Preliminary**: significant gain in merged signal efficiency @ same background rejection wrt cut-based analysis
- **Future**: try Recurrent Neural Networks to allow dynamic number of inputs and expand to resolved analysis.

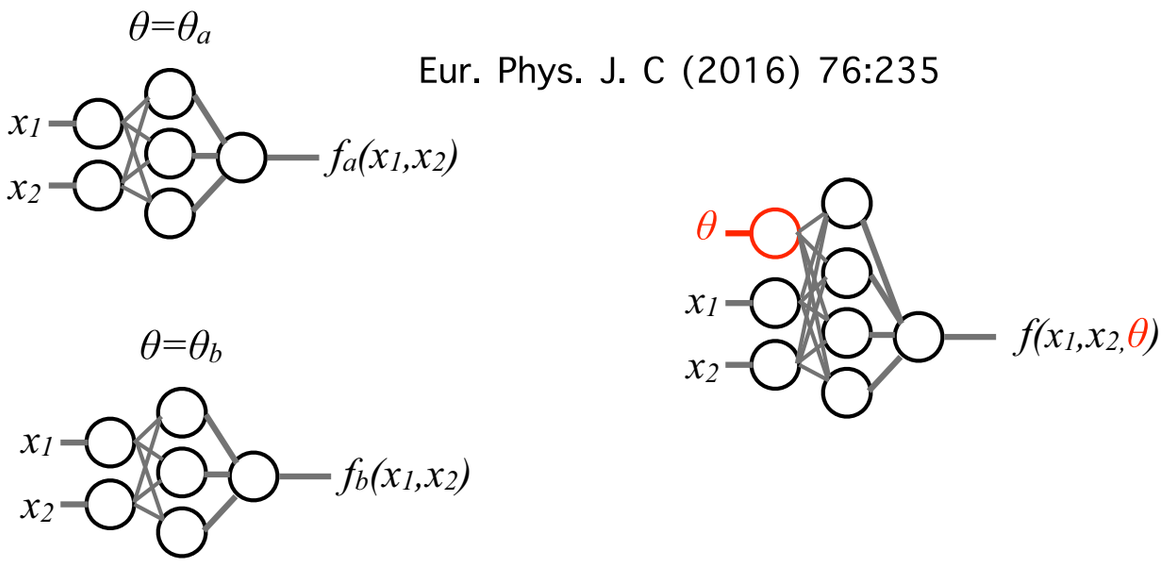


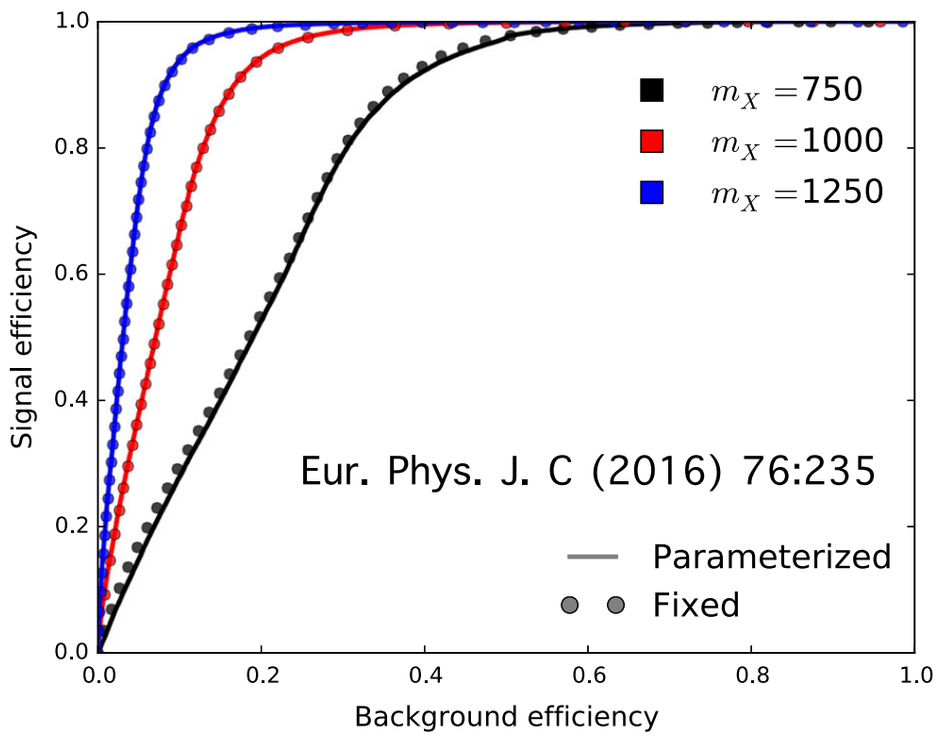
Input Variables	Leptons pt, E	Leptons eta, phi	Fatjet pt, E	Fatjet eta, phi	Fatjet D2,C2	Z(ll) mass, pt	MET
Set 0	x	x	x	x	x	x	x
Set 1	x		x			x	x
Set 2	x		x				

Preselection applied
Number of leptons = 2 (ee or μμ)
Number of fat jets >= 1 (highest pT jet selected)

Parameterized DNNs for high-energy physics

- **Problem:** Since the mass of the resonance is unknown how to best train a DNN when we want to interpret the observed data with a hypothetical signal at many different mass points? Avoid discontinuities in limit setting.
- Using a single parameterized DNN which tackles the full set of related tasks (**based on idea in Eur. Phys. J. C (2016) 76:235**)
  - Include as input feature one or more parameters that describe the larger scope of the problem (e.g new particle's mass)
- A parameterized classifier can smoothly interpolate between masses and replace sets of classifiers trained at individual values.
- **Benchmark in llqq analysis already seems promising**: Comparison of a parameterized DNN trained on ALL mass points except a single point and a DNN trained on ALL points yields same results.





Summary

- Working since mid October 2017 at Lecce on ML techniques for physics analysis
- Current research focused around the use of DNNs in ATLAS searches for heavy resonances decaying to dibosons
    1. Signal and background event classification with Deep NN to maximize sensitivity of cut-based analysis
    2. Development of parameterized DNNs to interpolate the DNN results across the mass range and ease the interpretation of observed data to several different hypothetical signals
- Results up to now are very encouraging although preliminary.
- Future goals:
    - Completely cover the llqq analysis using a DNN approach
    - Expand to different NN architectures depending on the problem (e.g RNN, Adversarial NN)
    - If time allows look into anomaly detection techniques for general BSM searches.

Fast parameterised simulation option in LHCb simulation framework

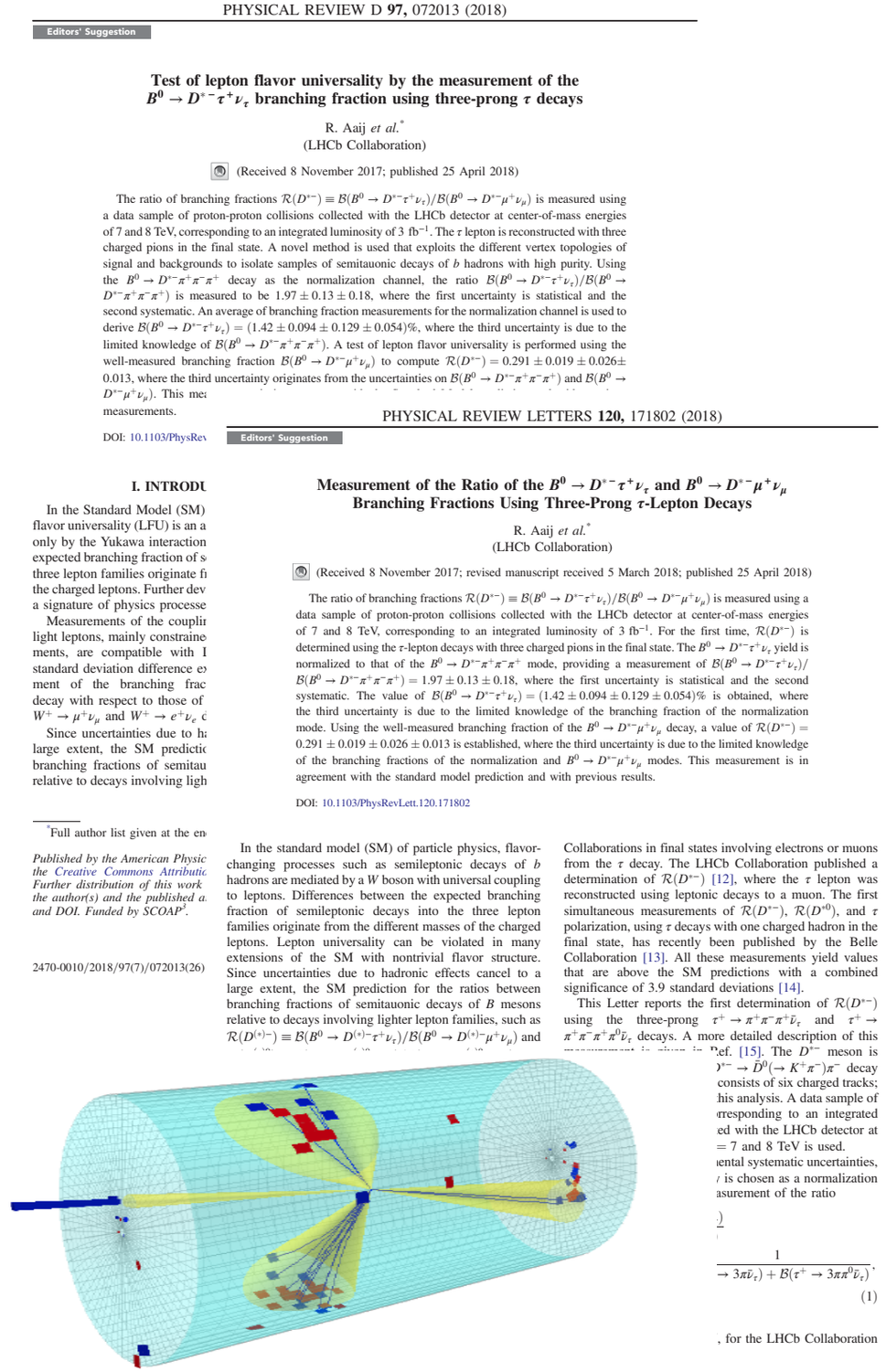
Benedetto Gianluca Siddi

On behalf of the LHCb Collaboration

Università degli studi di Ferrara
INFN - Sezione Ferrara

Who am I?

- Benedetto Gianluca Siddi
- Started my PhD in Ferrara in 2015 working on Lepton Flavour Universality Tests
  - Topic of my PhD thesis
  - Two articles (PRD and PRL) came out from the analysis
- In the meantime started to work on the implementation of a Parametrized Fast Simulation option to be integrated in the LHCb Simulation framework
- PhD thesis defended in February 2018
- INFN Fellow in scientific computing in Ferrara from January 2018
  - Continuing the work on FastSimulation and future upgrades for LHCb and simulation part of TimeSpot project



PHYSICAL REVIEW D **97,** 072013 (2018)

Editors' Suggestion

Test of lepton flavor universality by the measurement of the $B^0 \to D^{*-}\tau^+\nu_\tau$ branching fraction using three-prong τ decays

R. Aaij *et al.*[*]
(LHCb Collaboration)

(Received 8 November 2017; published 25 April 2018)

The ratio of branching fractions $\mathcal{R}(D^{*-}) \equiv \mathcal{B}(B^0 \to D^{*-}\tau^+\nu_\tau)/\mathcal{B}(B^0 \to D^{*-}\mu^+\nu_\mu)$ is measured using a data sample of proton-proton collisions collected with the LHCb detector at center-of-mass energies of 7 and 8 TeV, corresponding to an integrated luminosity of 3 fb^{-1}. The τ lepton is reconstructed with three charged pions in the final state. A novel method is used that exploits the different vertex topologies of signal and backgrounds to isolate samples of semitauonic decays of b hadrons with high purity. Using the $B^0 \to D^{*-}\pi^+\pi^-\pi^+$ decay as the normalization channel, the ratio $\mathcal{B}(B^0 \to D^{*-}\tau^+\nu_\tau)/\mathcal{B}(B^0 \to D^{*-}\pi^+\pi^-\pi^+)$ is measured to be $1.97 \pm 0.13 \pm 0.18$, where the first uncertainty is statistical and the second systematic. An average of branching fraction measurements for the normalization channel is used to derive $\mathcal{B}(B^0 \to D^{*-}\tau^+\nu_\tau) = (1.42 \pm 0.094 \pm 0.129 \pm 0.054)\%$, where the third uncertainty is due to the limited knowledge of $\mathcal{B}(B^0 \to D^{*-}\pi^+\pi^-\pi^+)$. A test of lepton flavor universality is performed using the well-measured branching fraction $\mathcal{B}(B^0 \to D^{*-}\mu^+\nu_\mu)$ to compute $\mathcal{R}(D^{*-}) = 0.291 \pm 0.019 \pm 0.026 \pm 0.013$, where the third uncertainty originates from the uncertainties on $\mathcal{B}(B^0 \to D^{*-}\pi^+\pi^-\pi^+)$ and $\mathcal{B}(B^0 \to D^{*-}\mu^+\nu_\mu)$. This mea… measurements.

DOI: 10.1103/PhysRev…

I. INTRODU…

In the Standard Model (SM) flavor universality (LFU) is an a… only by the Yukawa interaction… expected branching fraction of s… three lepton families originate fi… the charged leptons. Further dev… a signature of physics processe…

Measurements of the couplir… light leptons, mainly constraine… ments, are compatible with I… standard deviation difference e… ment of the branching frac… decay with respect to those of… $W^+ \to \mu^+\nu_\mu$ and $W^+ \to e^+\nu_e$ …

Since uncertainties due to h… large extent, the SM predictio… branching fractions of semitau… relative to decays involving ligh…

PHYSICAL REVIEW LETTERS **120,** 171802 (2018)

Editors' Suggestion

Measurement of the Ratio of the $B^0 \to D^{*-}\tau^+\nu_\tau$ and $B^0 \to D^{*-}\mu^+\nu_\mu$ Branching Fractions Using Three-Prong τ-Lepton Decays

R. Aaij *et al.*[*]
(LHCb Collaboration)

(Received 8 November 2017; revised manuscript received 5 March 2018; published 25 April 2018)

The ratio of branching fractions $\mathcal{R}(D^{*-}) \equiv \mathcal{B}(B^0 \to D^{*-}\tau^+\nu_\tau)/\mathcal{B}(B^0 \to D^{*-}\mu^+\nu_\mu)$ is measured using a data sample of proton-proton collisions collected with the LHCb detector at center-of-mass energies of 7 and 8 TeV, corresponding to an integrated luminosity of 3 fb^{-1}. For the first time, $\mathcal{R}(D^{*-})$ is determined using the τ-lepton decays with three charged pions in the final state. The $B^0 \to D^{*-}\tau^+\nu_\tau$ yield is normalized to that of the $B^0 \to D^{*-}\pi^+\pi^-\pi^+$ mode, providing a measurement of $\mathcal{B}(B^0 \to D^{*-}\tau^+\nu_\tau)/\mathcal{B}(B^0 \to D^{*-}\pi^+\pi^-\pi^+) = 1.97 \pm 0.13 \pm 0.18$, where the first uncertainty is statistical and the second systematic. The value of $\mathcal{B}(B^0 \to D^{*-}\tau^+\nu_\tau) = (1.42 \pm 0.094 \pm 0.129 \pm 0.054)\%$ is obtained, where the third uncertainty is due to the limited knowledge of the branching fraction of the normalization mode. Using the well-measured branching fraction of the $B^0 \to D^{*-}\mu^+\nu_\mu$ decay, a value of $\mathcal{R}(D^{*-}) = 0.291 \pm 0.019 \pm 0.026 \pm 0.013$ is established, where the third uncertainty is due to the limited knowledge of the branching fractions of the normalization and $B^0 \to D^{*-}\mu^+\nu_\mu$ modes. This measurement is in agreement with the standard model prediction and with previous results.

DOI: 10.1103/PhysRevLett.120.171802

[*]Full author list given at the en…

Published by the American Physic… the Creative Commons Attributio… Further distribution of this work… the author(s) and the published a… and DOI. Funded by SCOAP³.

2470-0010/2018/97(7)/072013(26)

In the standard model (SM) of particle physics, flavor-changing processes such as semileptonic decays of b hadrons are mediated by a W boson with universal coupling to leptons. Differences between the expected branching fraction of semileptonic decays into the three lepton families originate from the different masses of the charged leptons. Lepton universality can be violated in many extensions of the SM with nontrivial flavor structure. Since uncertainties due to hadronic effects cancel to a large extent, the SM prediction for the ratios between branching fractions of semitauonic decays of B mesons relative to decays involving lighter lepton families, such as $\mathcal{R}(D^{(*)-}) \equiv \mathcal{B}(B^0 \to D^{(*)-}\tau^+\nu_\tau)/\mathcal{B}(B^0 \to D^{(*)-}\mu^+\nu_\mu)$ and …

Collaborations in final states involving electrons or muons from the τ decay. The LHCb Collaboration published a determination of $\mathcal{R}(D^{*-})$ [12], where the τ lepton was reconstructed using leptonic decays to a muon. The first simultaneous measurements of $\mathcal{R}(D^{*-})$, $\mathcal{R}(D^{*0})$, and τ polarization, using τ decays with one charged hadron in the final state, has recently been published by the Belle Collaboration [13]. All these measurements yield values that are above the SM predictions with a combined significance of 3.9 standard deviations [14].

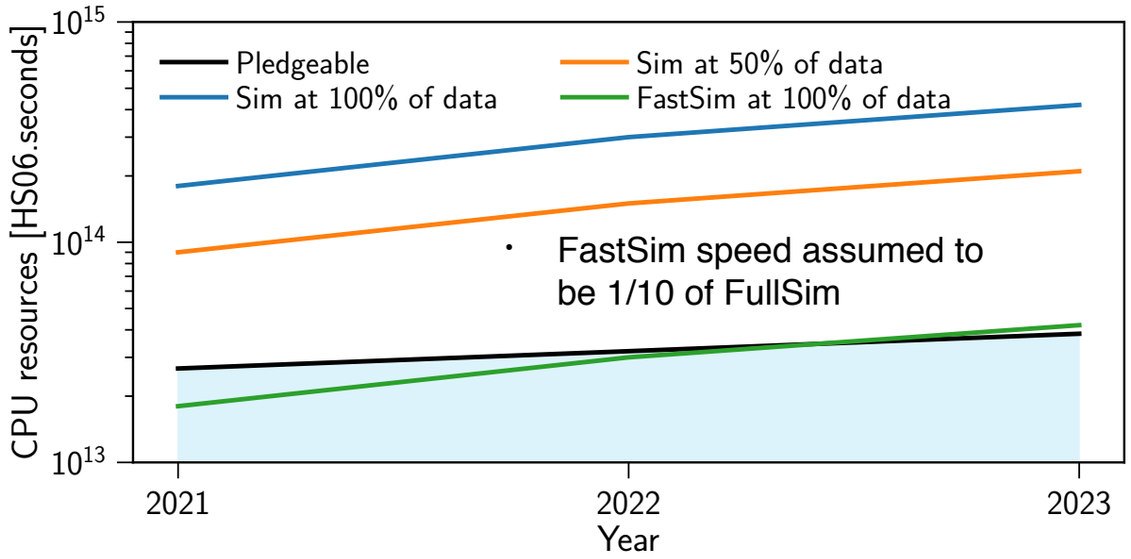
This Letter reports the first determination of $\mathcal{R}(D^{*-})$ using the three-prong $\tau^+ \to \pi^+\pi^-\pi^+\bar{\nu}_\tau$ and $\tau^+ \to \pi^+\pi^-\pi^+\pi^0\bar{\nu}_\tau$ decays. A more detailed description of this measurement is given in Ref. [15]. The D^{*-} meson is … $D^{*-} \to \bar{D}^0(\to K^+\pi^-)\pi^-$ decay … consists of six charged tracks; … his analysis. A data sample of … rresponding to an integrated … ed with the LHCb detector at … = 7 and 8 TeV is used. …ental systematic uncertainties, … is chosen as a normalization … asurement of the ratio

$$\frac{1}{\mathcal{B}(\tau^+ \to 3\pi\bar{\nu}_\tau) + \mathcal{B}(\tau^+ \to 3\pi\pi^0\bar{\nu}_\tau)}, \quad (1)$$

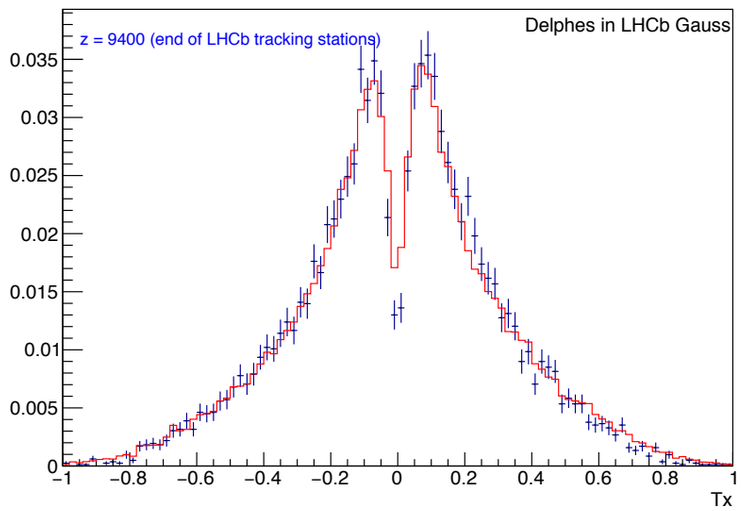
…, for the LHCb Collaboration

FastMC: Integration of Delphes in LHCb simulation framework

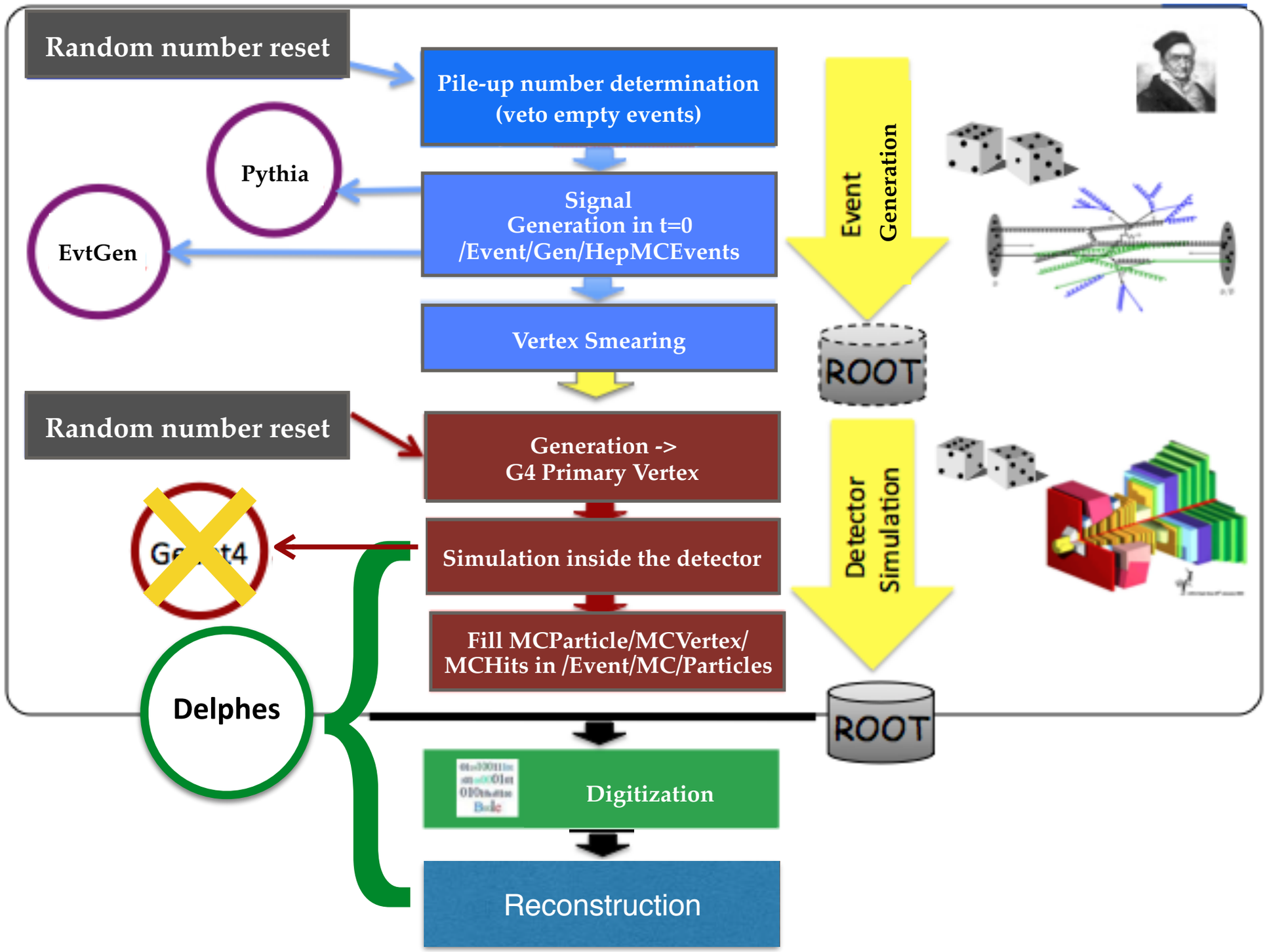
- The role of Monte Carlo simulation in high energy physics experiment is to mimic the behaviour of a detector to understand experimental conditions and performance
- Systematics uncertainties in most of the analysis are dominated by the MC
- Large MC samples → large resources
- New simulation options needs to be investigated



- Requirements for a FastMC:
  - Two orders of magnitude faster than GEANT4
  - Less CPU consuming
  - Reconstructed particle information in order to use the standard LHCb tools for analysis
  - As close as possible to the full simulation

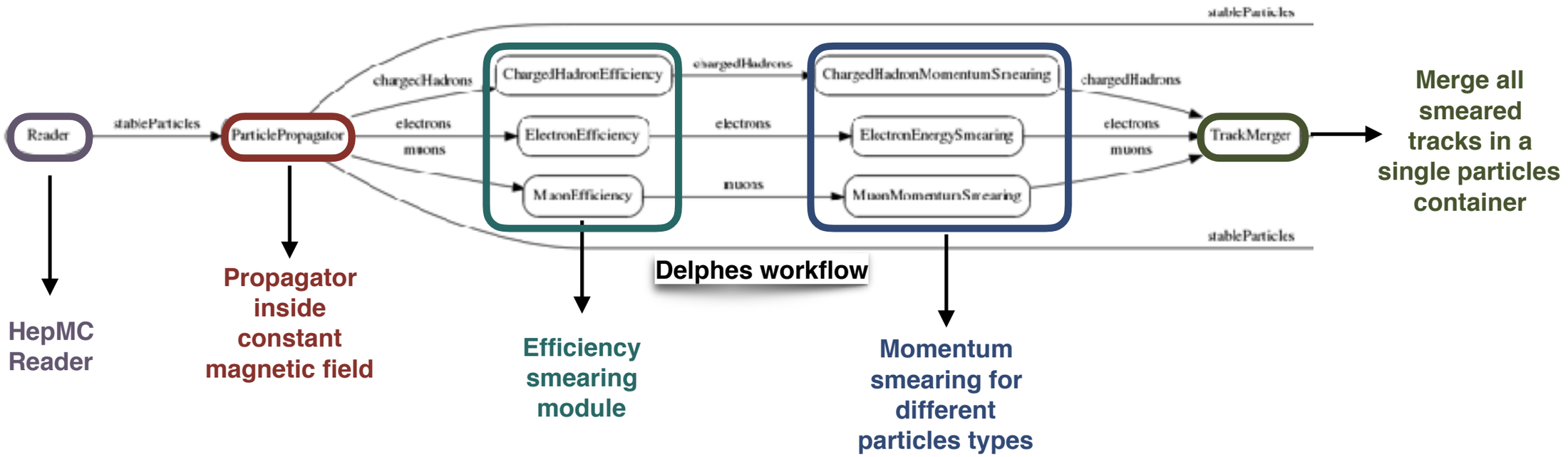


Fast simulation with Delphes in LHCb



Delphes

- Delphes + modifications for LHCb, has been integrated in LHCb simulation framework Gauss.
  - It takes in input particles generated from the generator part of Gauss,
  - It writes as output objects in the format necessary for LHCb analysis framework.
- No lower level reconstructed objects!

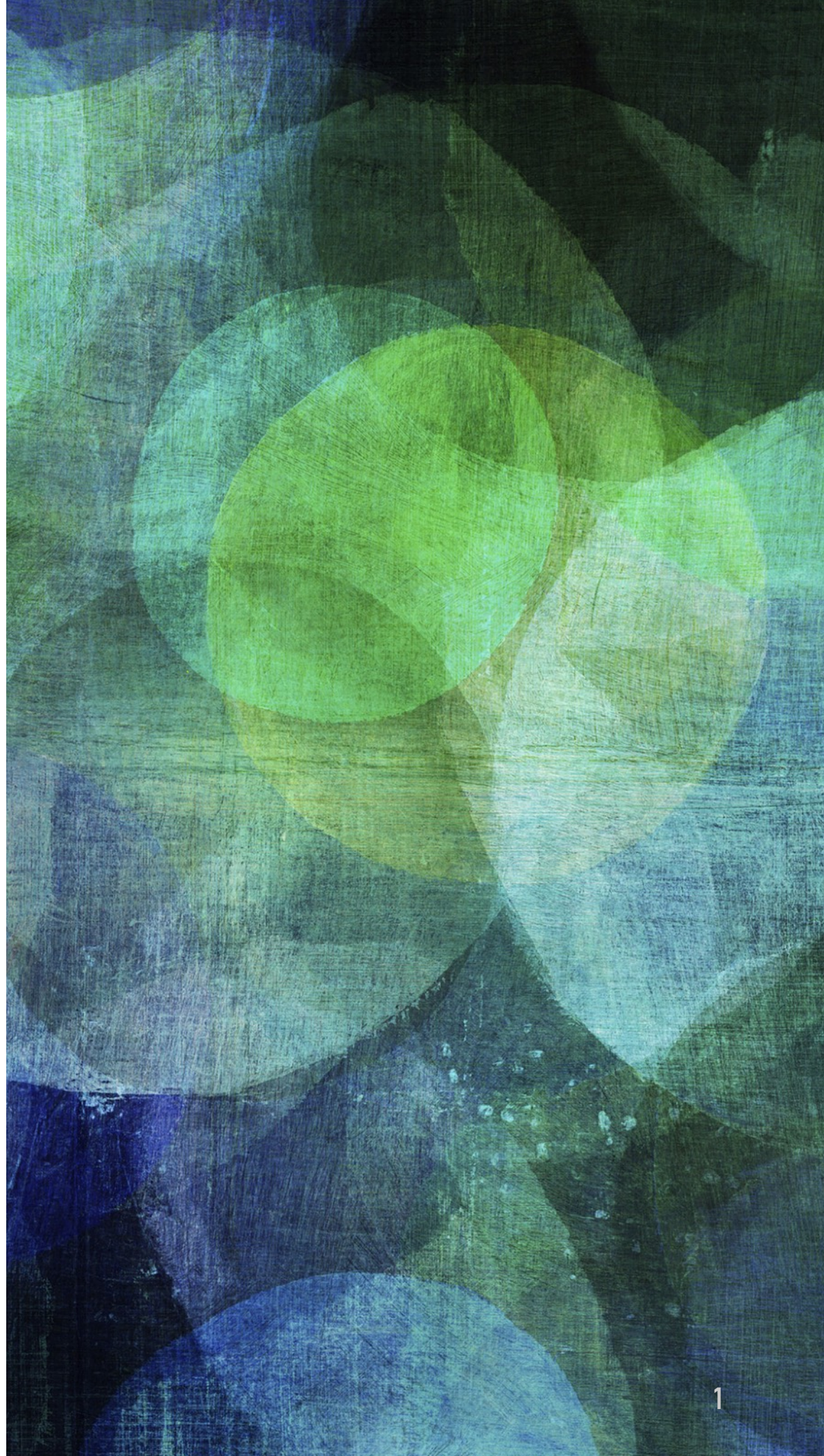


- Working on implementing relevant quantities of reconstructed tracks, e.g., covariance matrices, fit χ^2, ghost probabilities
- Work to be done :
  - Particle Identification probabilities, calorimeter response for charged and neutral particles.
  - Finalize the output of the objects filled with the information needed to be used in the LHCb analysis framework in order to perform physics analyses
  - Review the code in order to make it thread safe and multithreading to be used in the new LHCb Framework.

EXPLOITING PARALLELIZABLE ARCHITECTURES AND ARTIFICIAL INTELLIGENCE AT THE LHC EXPERIMENTS

Cesare Calabria

"INFN and The Future of Scientific Computing
Episode I: The HPC Opportunity"
2018/05/04



PAST ACTIVITIES AND RESPONSIBILITIES

- **Research activity within the CMS experiment**
  - **Analysis of the date collected with the CMS experiment at LHC**
    - Electroweak physics: measurement of the Z boson production cross section through its decay to a tau lepton pair
    - Search for a Standard Model Higgs boson produced in association with a W vector boson
  - **Study of the physics objects in CMS**
    - Study of the muon reconstruction and identification performance in CMS
    - Study of the hadronic tau reconstruction and identification performance in CMS
  - **Activities concerning the detectors**
    - Study of the RPC performance in CMS
    - Upgrade of the CMS Muon System forward region with new detectors based on GEM technology
    - Phase-2 upgrade of the Muon System

- **Responsibilities within the CMS experiment**
  - 2010 - 2012: Responsible for the study and monitoring of the RPC efficiency (L3)
  - **2014 - 2015: GEM Reconstruction and Validation Coordinator (L3)**
  - **2014 - 2017: GEM Software and Online Contact for Upgrade (L3)**
  - 2015 - 2017: GEM DPG Coordinator (L2)
  - **2016 - 2017: Muon Phase-II Simulation Coordinator (L3)**
  - **2016 - 2017: Contact person between Upgrade Studies Group and CMS Offline & Computing group (L3)**
  - 2016 - 2017: Link person CMS - Bari Tier2

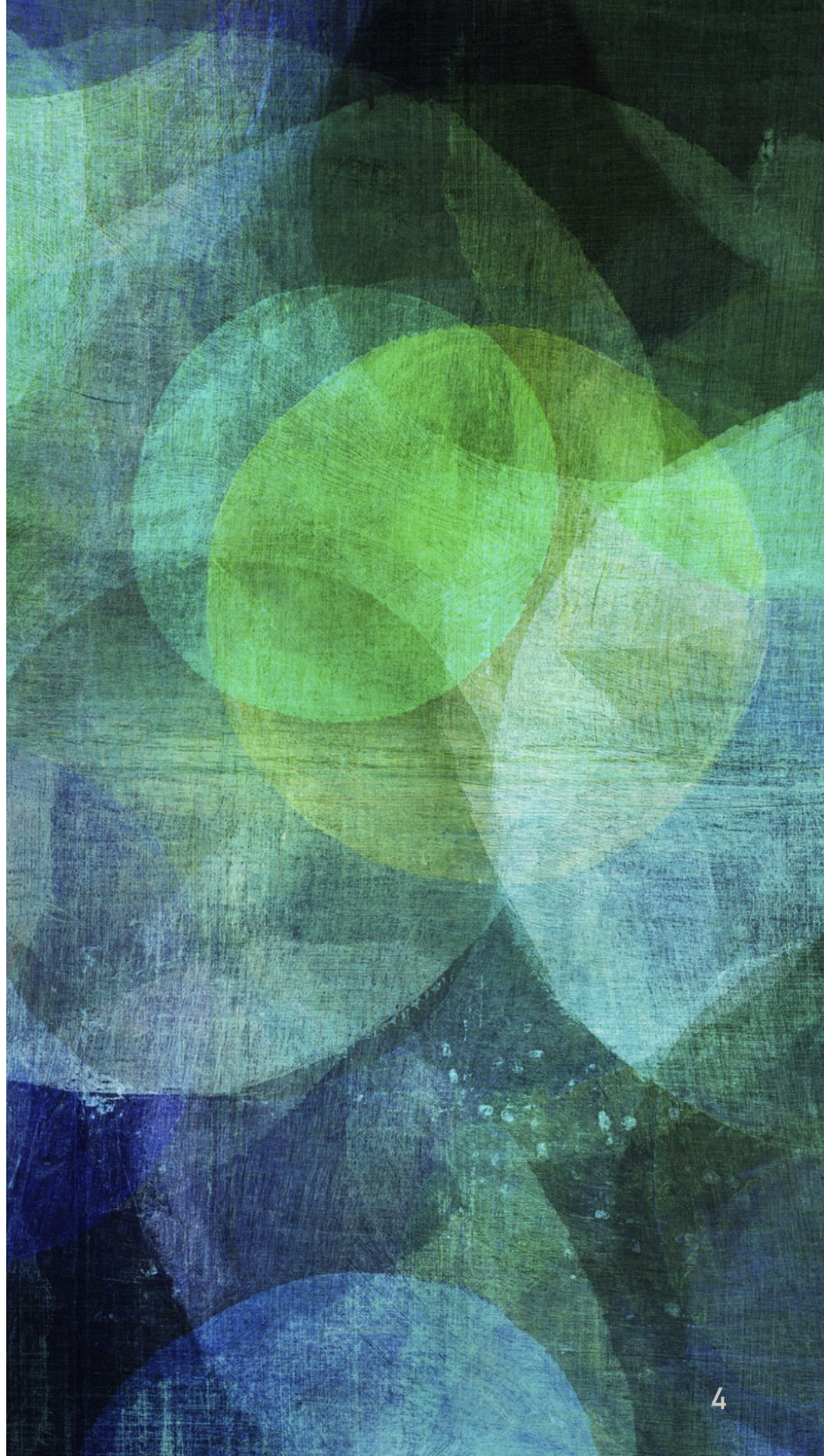
- **Development and integration of the software needed for the Phase2 Muon Upgrade studies (muon reconstruction and identification, validation tools, study of the neutron background and muon perfomance…)**

CURRENT ACTIVITIES

In the context of my INFN "fellowship" regarding the R&D on scientific computing for innovative solutions for the LHC experiments, my activity is twofold:

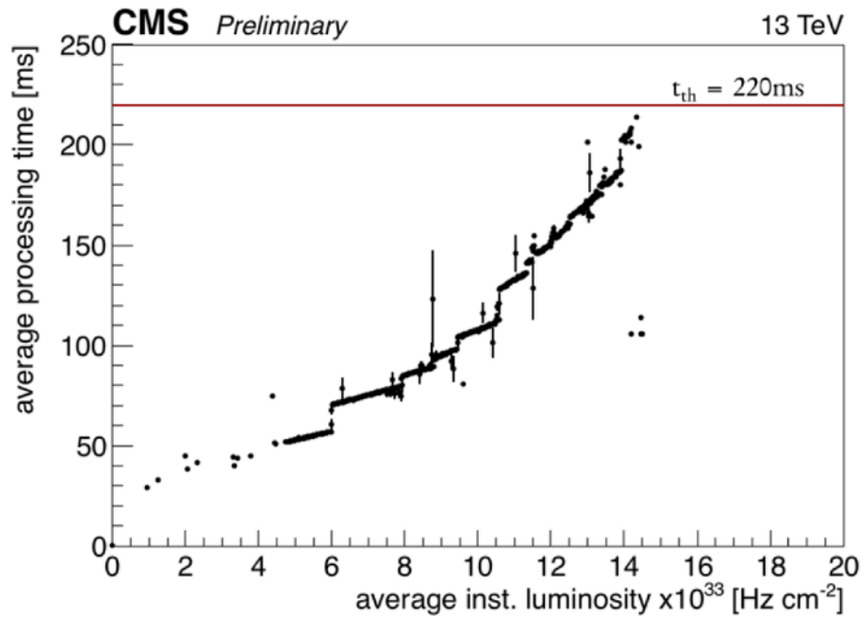
1. **Parallel programming on GPU: I am collaborating with the "Future tracking" group**
   - ➤ The group takes care in CMS of developing a demonstrator for the pixel tracking on GPU and all the other infrastructures needed to exploit at best the available hardware resources (heterogeneous computing)
   - ➤ **So far I contributed to the implementation on GPU and to the optimization of the first step of the chain: the unpacking of raw data (Raw2Digi) for the pixel detector (details in the next slides)**
2. **Machine learning application: I am collaborating with the "ML Muon" group**
   - ➤ The groups take care inside the CMS Muon community (DT, RPC, CSC, GEM) of developing innovative tools for monitoring the performance of the CMS Muon System and the detection of its anomalies
   - ➤ **Currently I am working on the development of a monitoring tool based on ML techniques for the DT Trigger System**
   - ➤ On long term this R&D work is meant to lead to a tool that can be run at different stages of the L1 Muon Trigger (details in the next slides)

CMS PIXEL TRACKING ON GPU



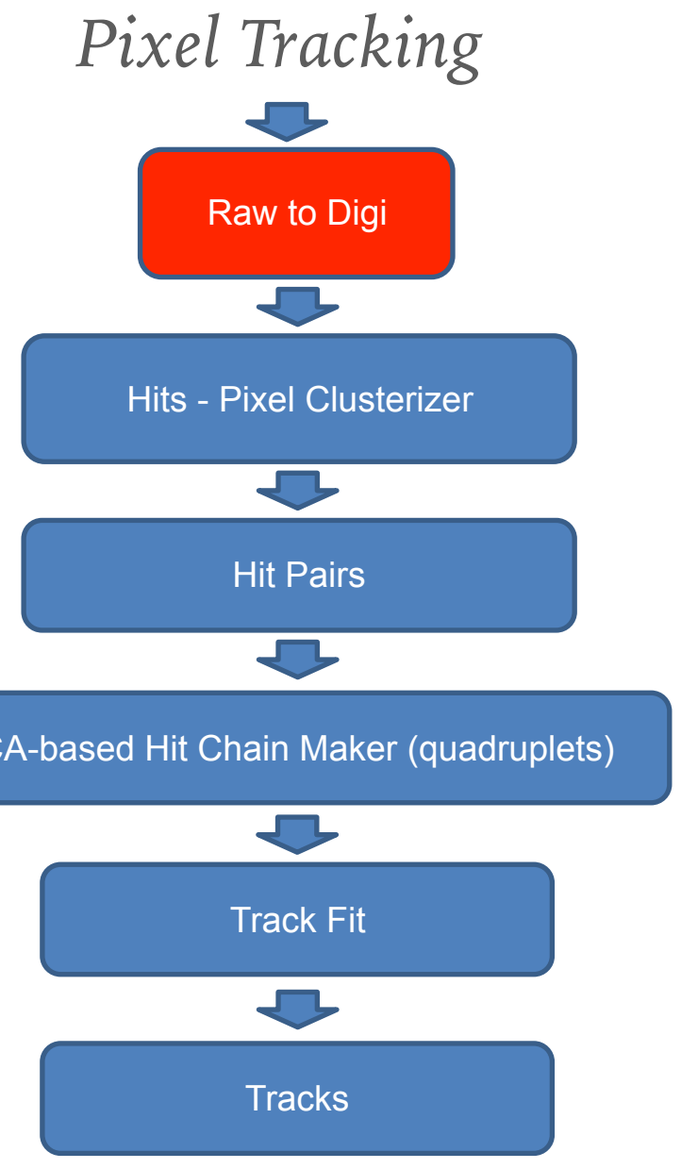
THE CMS TRIGGER SYSTEM

- **CMS Trigger System**
  - Reduce input rate (40 MHz) to a data rate (~1 kHz) that can be stored, reconstructed and analyzed Offline maximizing the physics reach of the experiment
  - **Level 1 Trigger**
    - Coarse readout of the Calorimeters and Muon detectors
    - Implemented in custom electronics (ASICs and FPGAs)
    - Output rate limited to 100 kHz by the readout electronics
  - **High Level Trigger**
    - Readout of the whole detector with full granularity
    - Output rate limited to an average of ~1 kHz by the Offline resources
    - **Today the CMS HLT online farm consists of ~22k Intel Xeon cores**
      - The current approach: one event per logical core
      - **Pixel Tracks cannot be reconstructed for all the events at the HLT**
      - This will be even more difficult at higher pile-up
        - Combinatorial time in pixel seeding O(pileup!) in worst case



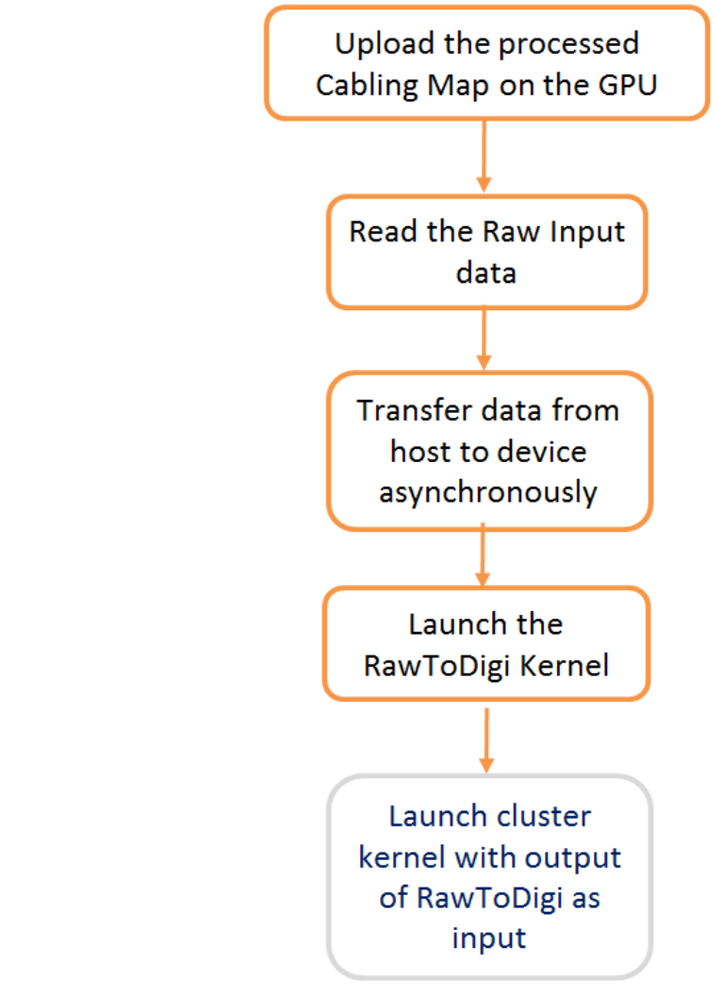
THE CMS PIXEL TRACKING AND THE PATATRACK PROJECT

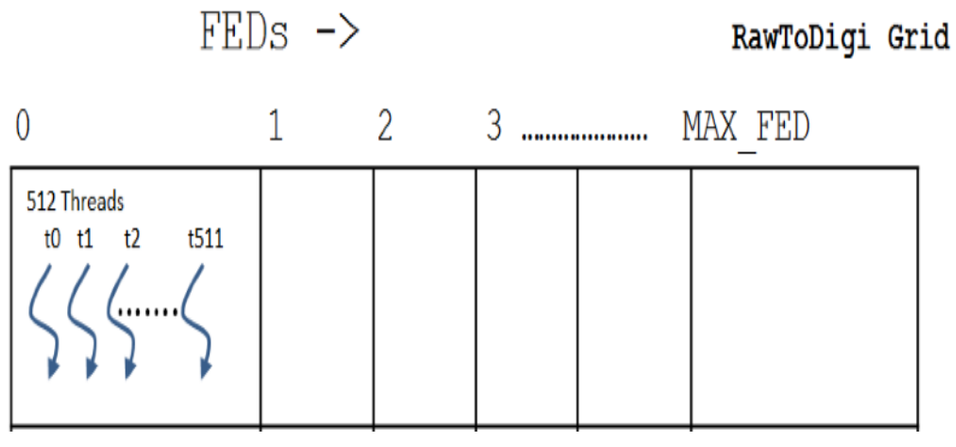
- **Solution (objective of the “Patatrack" project in CMS):**
  - **Develop a hybrid CPU-GPU application that takes RAW data coming from the pixel detector and gives Tracks as result**
  - Trigger average latency should stay within 220ms
  - GOAL: demonstrator ready by 09/2018 to run parasitically at the HLT farm (in order to be included for Run3 and then hopefully Run4)
- **Ingredients:**
  - Massive parallelism within the event
  - Avoid useless data transfers and transformations
  - Simple data formats optimized for parallel memory access
  - Renovation at algorithmic level
- **My contribution to the project is on the implementation on GPU of the first step of the pixel tracking chain: Raw2Digi step**



RAW2DIGI ON GPU: STATUS OF THE IMPLEMENTATION

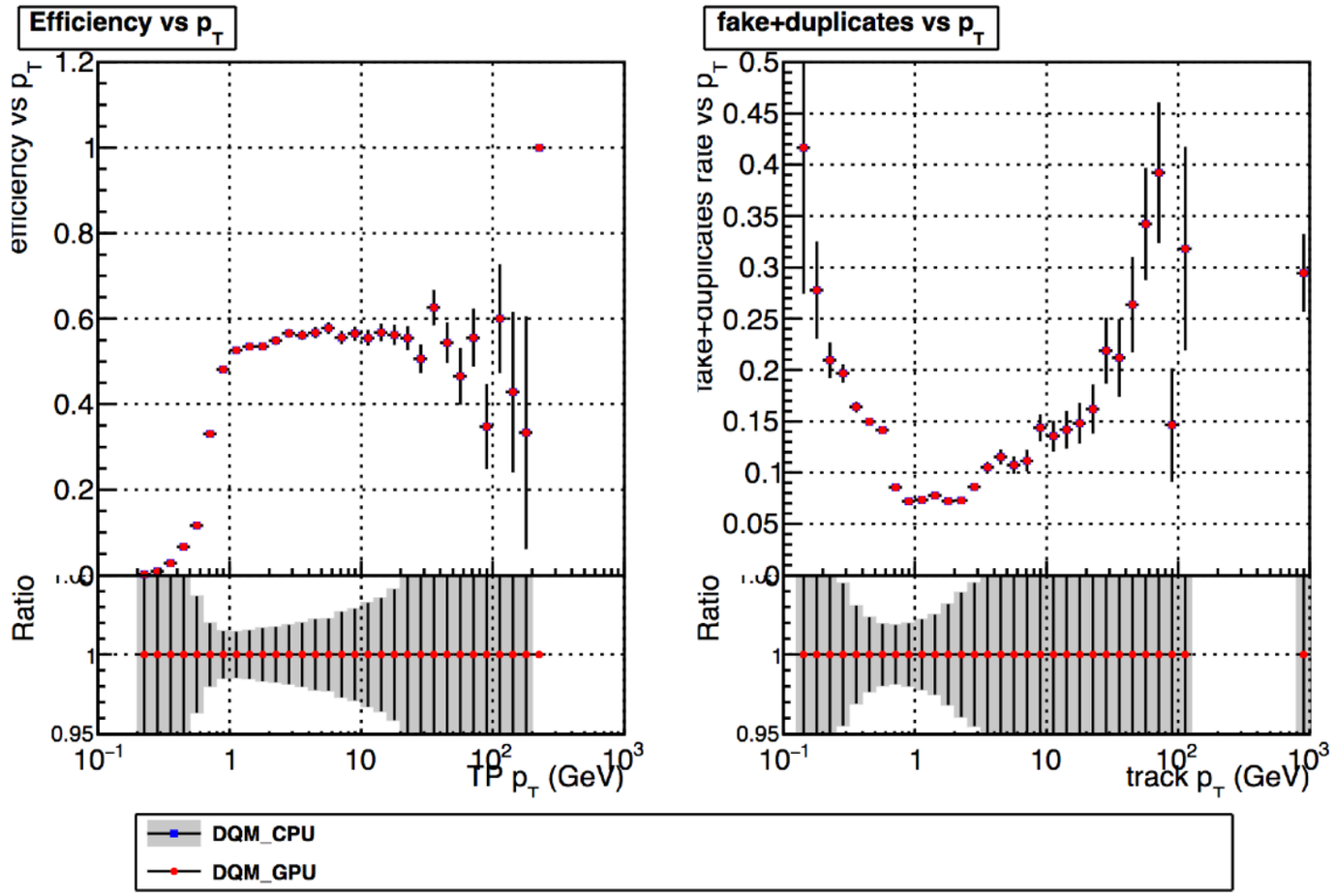
- **Main goal: reproduce what the CPU code does with a simpler and parallel implementation (CUDA) and try to speed up as much as possible the processing**
- **A fully working implementation of the pixel raw to digi algorithm on GPU is ready**
  - It unpacks x, y pixel coordinates and the corresponding adc count
  - It unpacks also FED errors
  - It is already integrated in the CMS framework analysis
- **How GPU parallel architecture is exploited**
  - Each FED is assigned to a block of threads
  - Words coming from a FED are saved in an array, copied to the GPU memory and assigned to the threads of the block where they are unpacked in parallel
  - Each thread executes the same set of instructions (kernel) on each word
- **Optimization of the memory usage and memory transfers for speeding up the algorithm (details in the backup slides)**



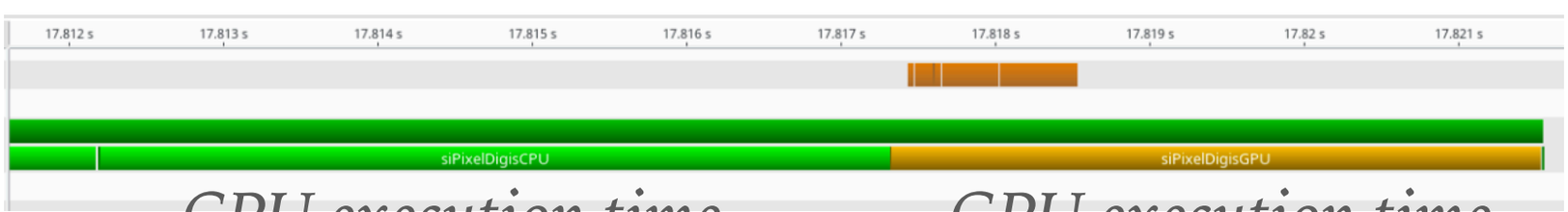


RAW2DIGI ON GPU: STATUS OF THE IMPLEMENTATION

- The GPU results have been validated against the CPU algorithm using the official validation plots
  - No differences between the two implementations
- We started to study the performance of the GPU with specific tool provided by NVIDIA
  - More optimization is possible
    - Reducing the time spent in the host-device and device-host memory copy
    - Optimizing the kernel
    - Maximizing the concurrency



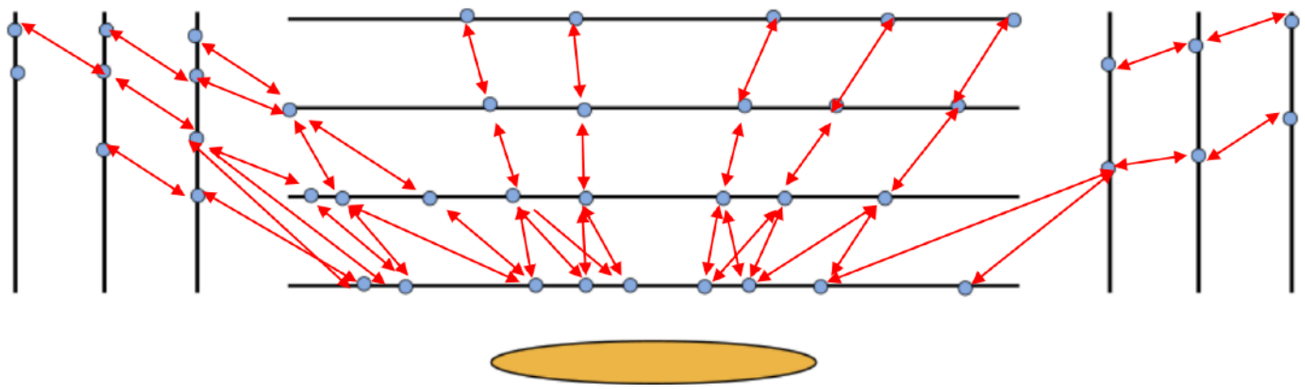
- We discovered almost all the time is spent on the CPU
  - We are also re-engineering the remaining part of the serial code (a.k.a. host code) to improve the performance and reduce the CPU execution time



CPU execution time *GPU execution time*

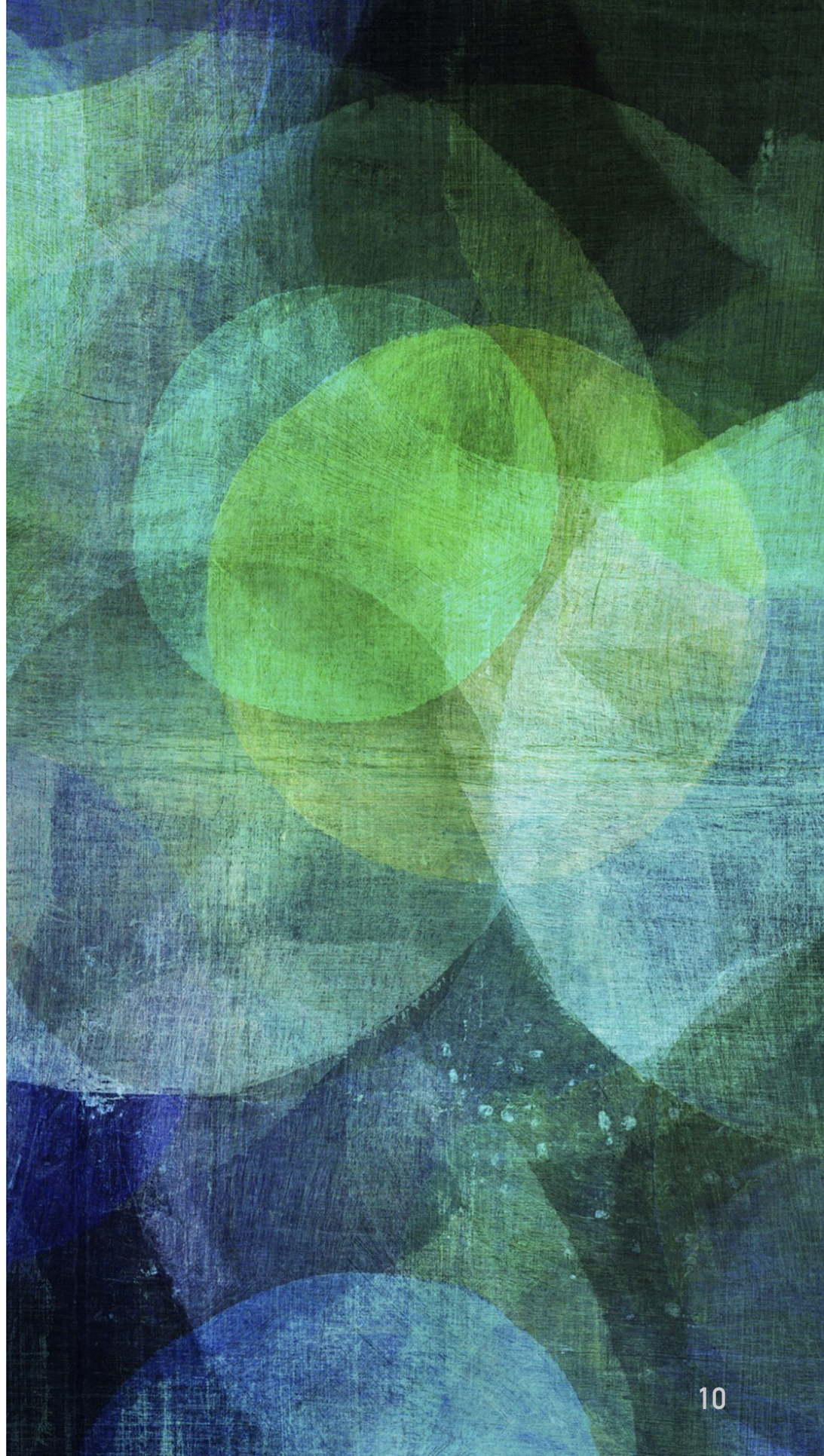
CONCLUSIONS AND NEXT STEPS

- **The work on the R2D step on GPU is completed**
- More refinements and improvements are possible, but those are left for the future
- **NEXT: develop a memory arena (in CUDA)**
  - A memory arena is simply a large, contiguous chunk of memory that is allocated once and then used to manage memory manually by handing out smaller chunks of that memory
  - This tool will be used to manage dynamically and efficiently the memory needed for saving the doublets used by the track seeding algorithm (for the quadruplet generation), since its number increases dramatically with PU



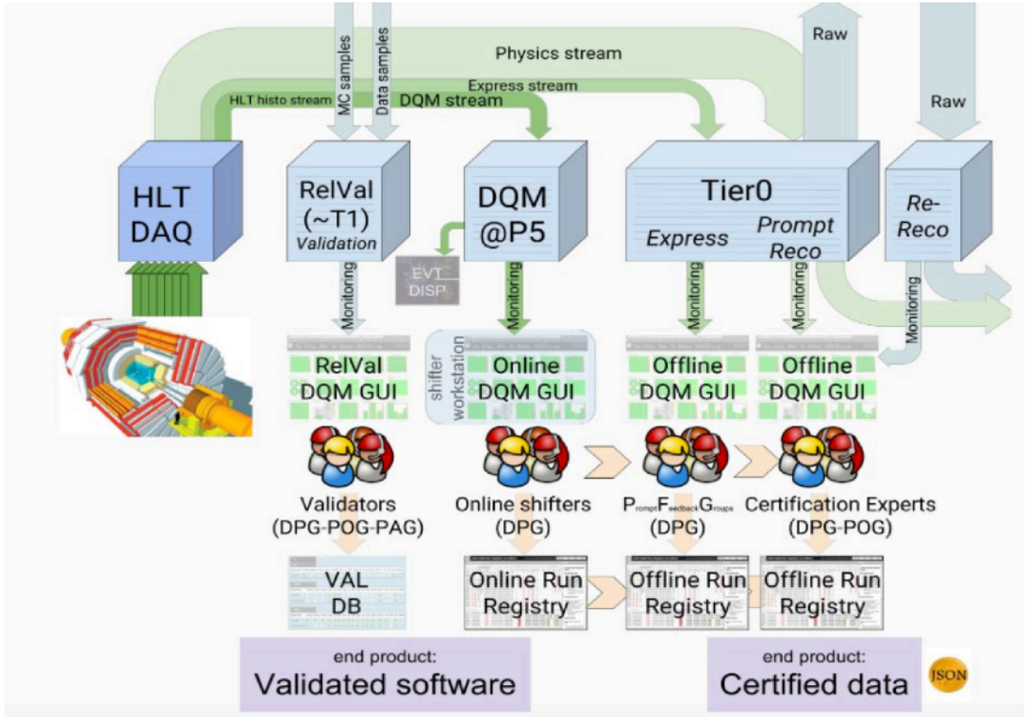
- **More infos about the Patatrack project at the link below:**
  - **https://patatrack.web.cern.ch/patatrack/**

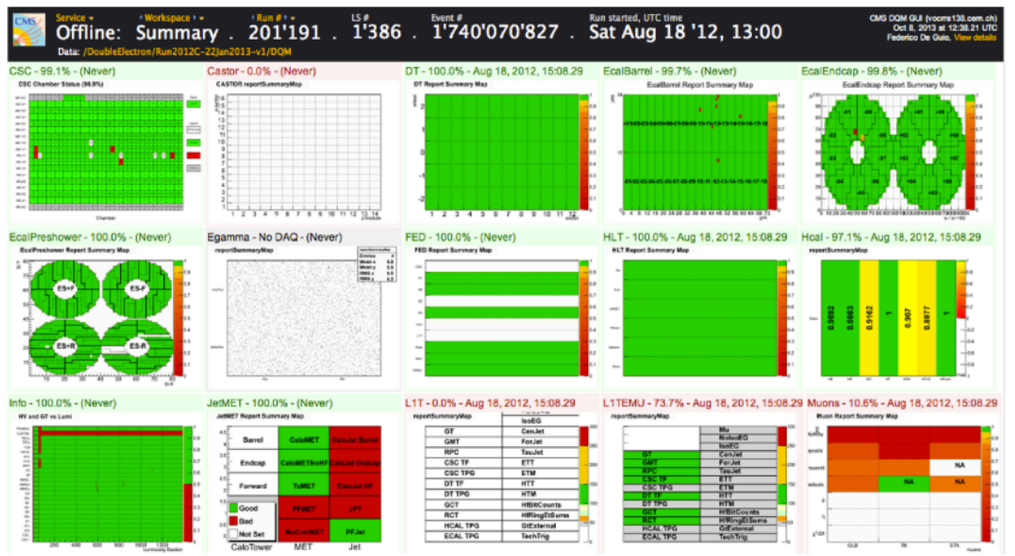
MACHINE LEARNING FOR THE CMS DATA QUALITY MONITORING



CMS DATA QUALITY MONITORING (DQM) SYSTEM

- **A critical asset to guarantee a high-quality data for physics analyses (online and offline)**
- **Online DQM assess data goodness and identifies emerging problems in the detector**
  - Data with poor quality is flagged by eyeballing DQM GUI and comparing a set of histograms to a reference good sample
- **Problems with current strategy:**
  - **Delay:** human intervention and tests require collecting sufficient statistics
  - **Volume budget:** amount of quantities a human can process in a finite time period
  - **Human driven decision process:** alarms based on shifter judgment
  - **Changing running conditions:** reference samples change over time
  - **Manpower:** the effort to train a shifter and maintain instructions

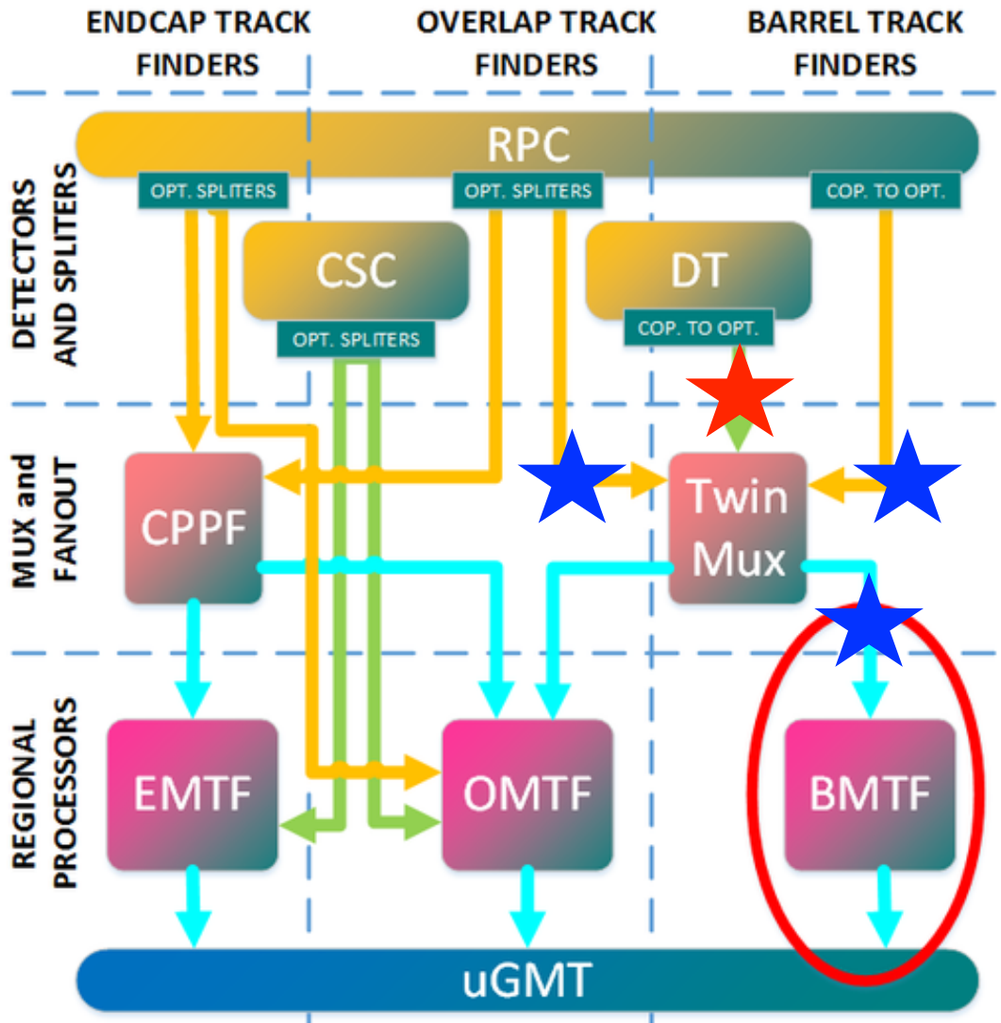




A TOOL FOR MONITORING THE L1 BARREL TRIGGER WITH ML

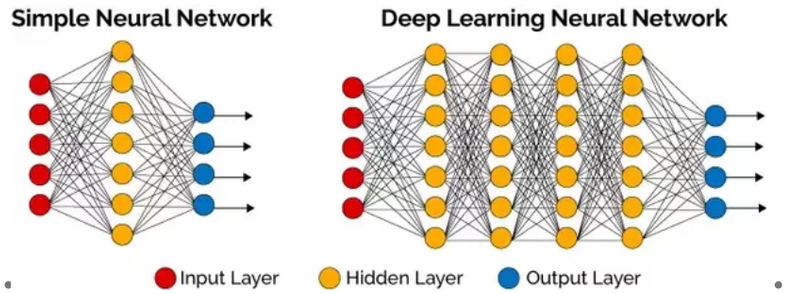
- **GOAL: Use ML/DL techniques for developing an innovative tool for the L1 Barrel Trigger rate monitoring in CMS**
  - To be run at the level of TwinMux inputs (DT and RPC inputs), TwinMux output and BMTF input
- The algorithm must:
  - correlate trigger rates and instantaneous luminosities coming from CMS database
  - identify chamber(s) with rate problem(s)
  - correlate different sources of information to make a diagnosis of the issue, e.g.:
    - all rates up to TwinMux output are in line with expectation for a given inst. lumi, but BMTF input is crazy ⇒ suspect communication issue between TwinMux Output and BMTF
- **Consumer: online operation teams**

Starting the project from the DT system!

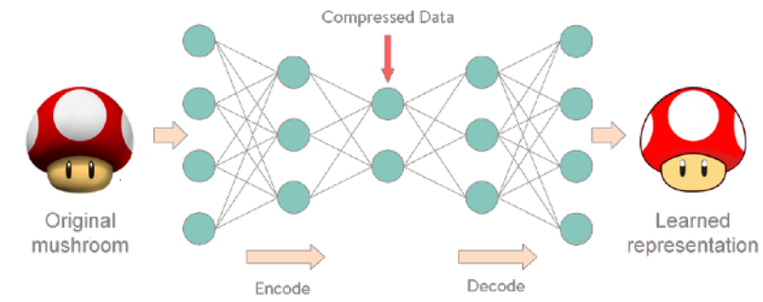


Details about the TwinMux and BMTF algorithm in the backup slides

ALGORITHMS



- **Input dimensionality: 10 features**
  - [system, wheel, sector, station, rate, rate uncertainty, inst. lumi., lumi/rate, uncertainty on ratio]
- **Building a deep neural network (DNN)**
  - Four hidden layers with 32 neurons
- **Trying an autoencoder (AE), i.e. a semi-supervised approach**
  - Only the sample with normalies is needed for the training
  - The network learns the features of good data in a processs of encoding-decoding
  - After that it should be able to re-reconstruct with some precision only the good data
  - Bottle neck leads to a dimensionality of 6 (from 10)
- **Testing on a DT known issue:**
  - Trigger board W+1, S4, MB3 is permanently off



Layer (type)	Output Shape	Param #
input_ann (Reshape)	(None, 10, 1)	0
flatten_ann (Flatten)	(None, 10)	0
dense_ann (Dense)	(None, 32)	352
dense_ann2 (Dense)	(None, 32)	1056
dense_ann3 (Dense)	(None, 32)	1056
dense_ann4 (Dense)	(None, 32)	1056
output_ann (Dense)	(None, 2)	66

Total params: 3,586
Trainable params: 3,586
Non-trainable params: 0

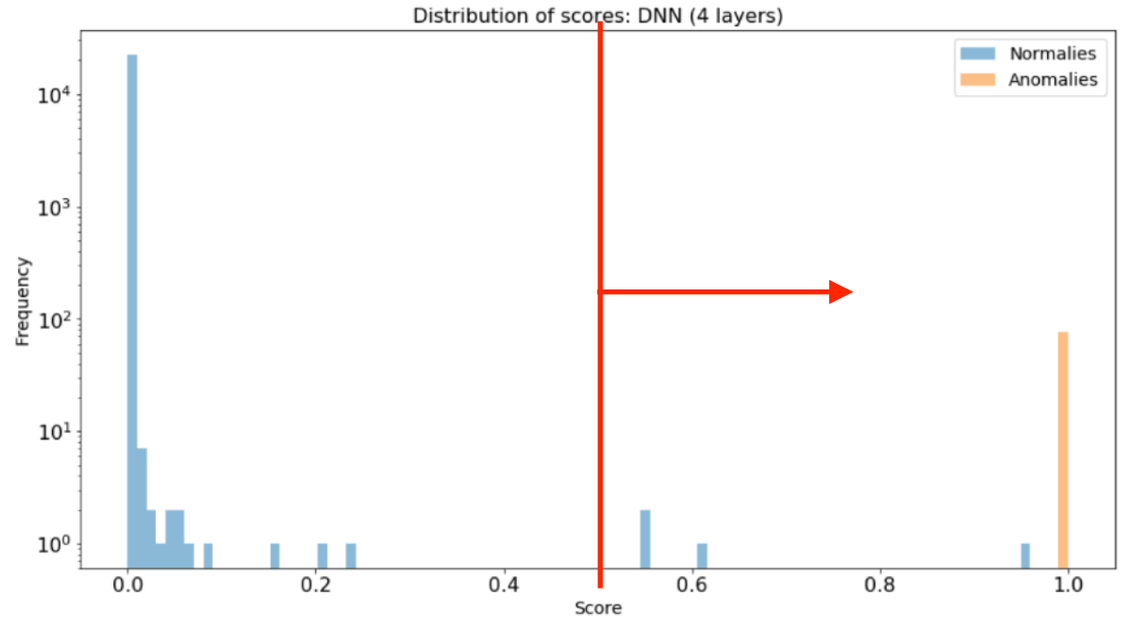
Autoencoder Architecture:

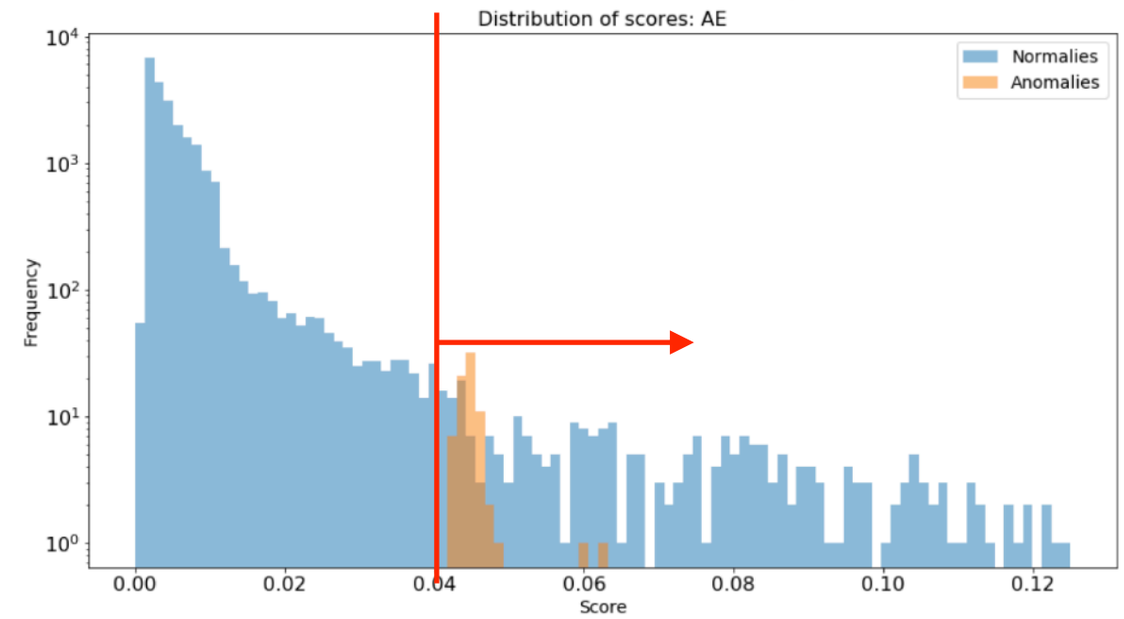
Layer (type)	Output Shape	Param #
input_29 (InputLayer)	(None, 10)	0
dense_281 (Dense)	(None, 10)	110
dense_282 (Dense)	(None, 9)	99
dense_283 (Dense)	(None, 8)	80
dense_284 (Dense)	(None, 7)	63
dense_285 (Dense)	(None, 6)	48
dense_286 (Dense)	(None, 6)	42
dense_287 (Dense)	(None, 7)	49
dense_288 (Dense)	(None, 8)	64
dense_289 (Dense)	(None, 9)	81
dense_290 (Dense)	(None, 10)	100

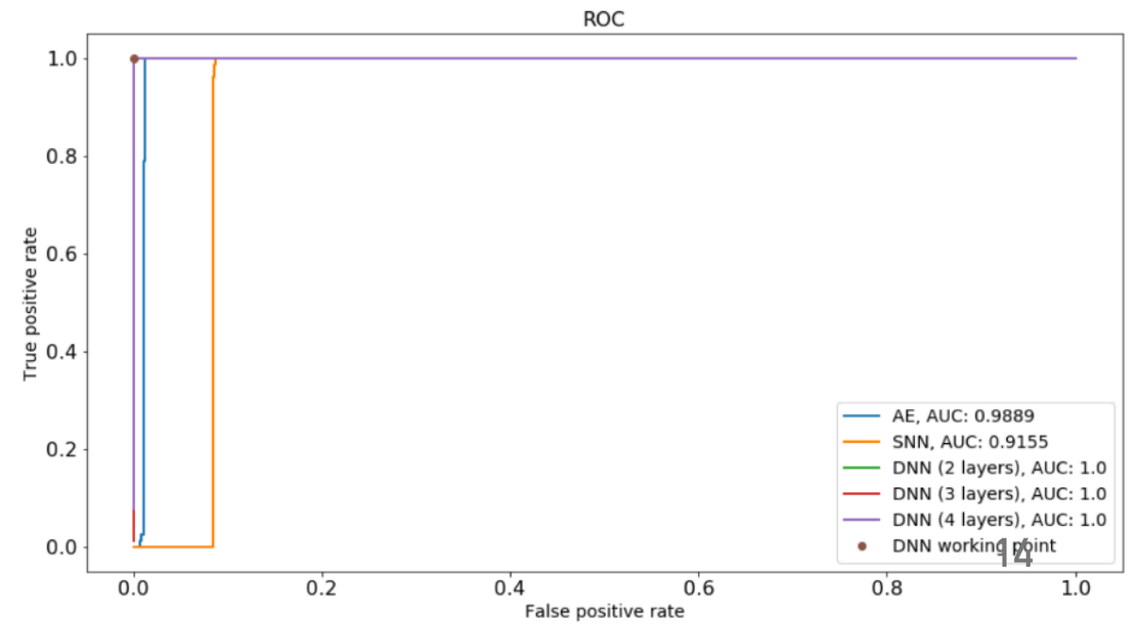
Total params: 736
Trainable params: 736
Non-trainable params: 0

PERFORMANCE

- **DNN shows very good separation between normalies and anomalies**
  - The DNN is able to classify in the correct way the normalies and anomalies in the test sample
  - Tried some options for the number of layers: good results are reached already with 2 layers
- **The classification with the AE is not clear as the DNN, anyway it is possible to fix a WP to recognize:**
  - 100% of the true positives with a 1% of false positives
  - AE approach seems promising:
    - no need to provide labels for anomalies
    - it can spot unforeseen problems
- Tried also some other classic approaches, but DNN and AE provide the best performance (see backup slides)

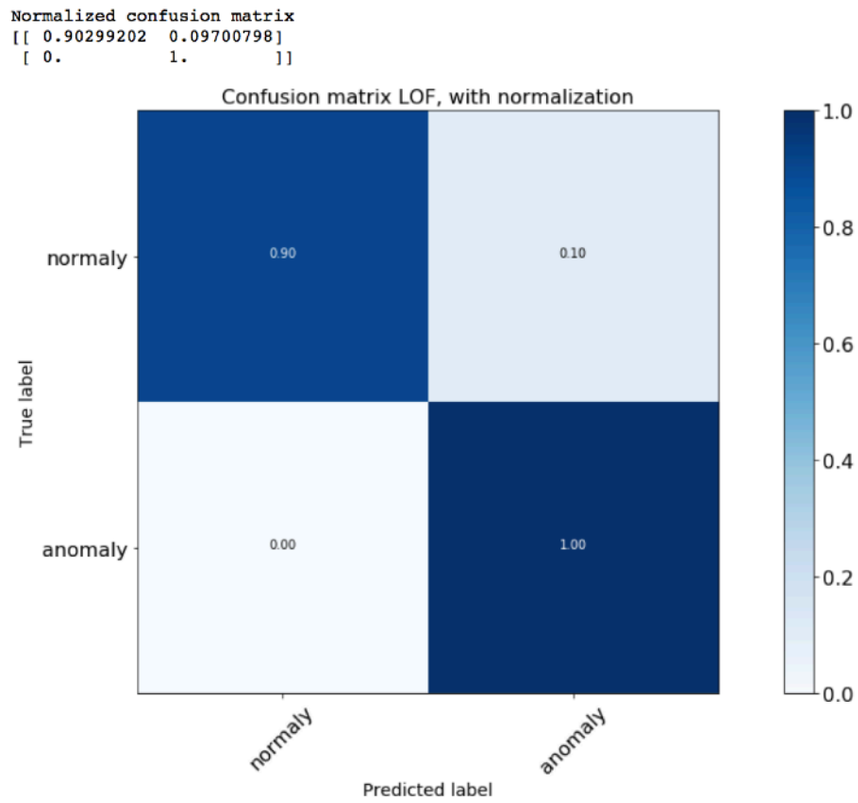


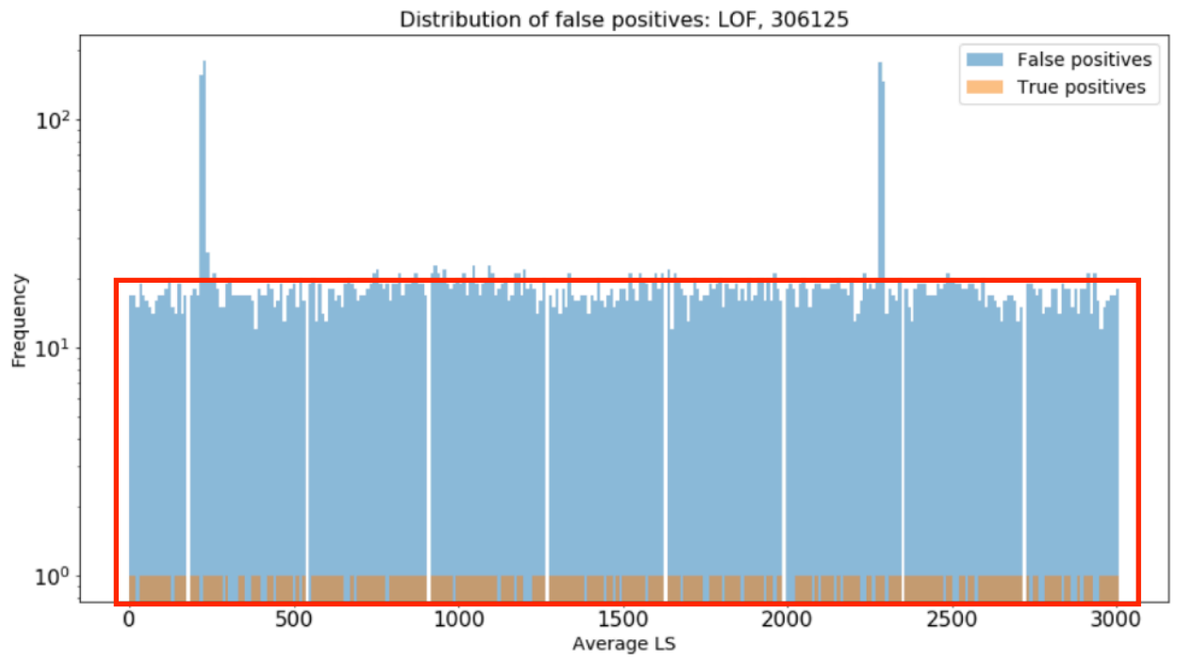




FURTHER ALGORITHMS: GOING COMPLETELY UNSUPERVISED

- **Moving to completely unsupervised algorithms is the best approach in order to detect all the possible anomalies without specific trainings in general performed over a limited number of anomalies that will never cover all the possible cases**
- **Local outlier factor (LOF)**
  - Based on k-NearestNeighbor algorithm
  - It detects all the known anomalies, but with too many false positives (10%):
    - < 2 FPs per LS, but still too much
  - Details here: http://scikit-learn.org/stable/modules/outlier_detection.html and in the backup slides
- **Looking into clustering algorithms (like K-Means clustering)**
  - Started working on it but the method and the results need to be better understood
  - Details here: http://scikit-learn.org/stable/modules/clustering.html#k-means and in the backup slides



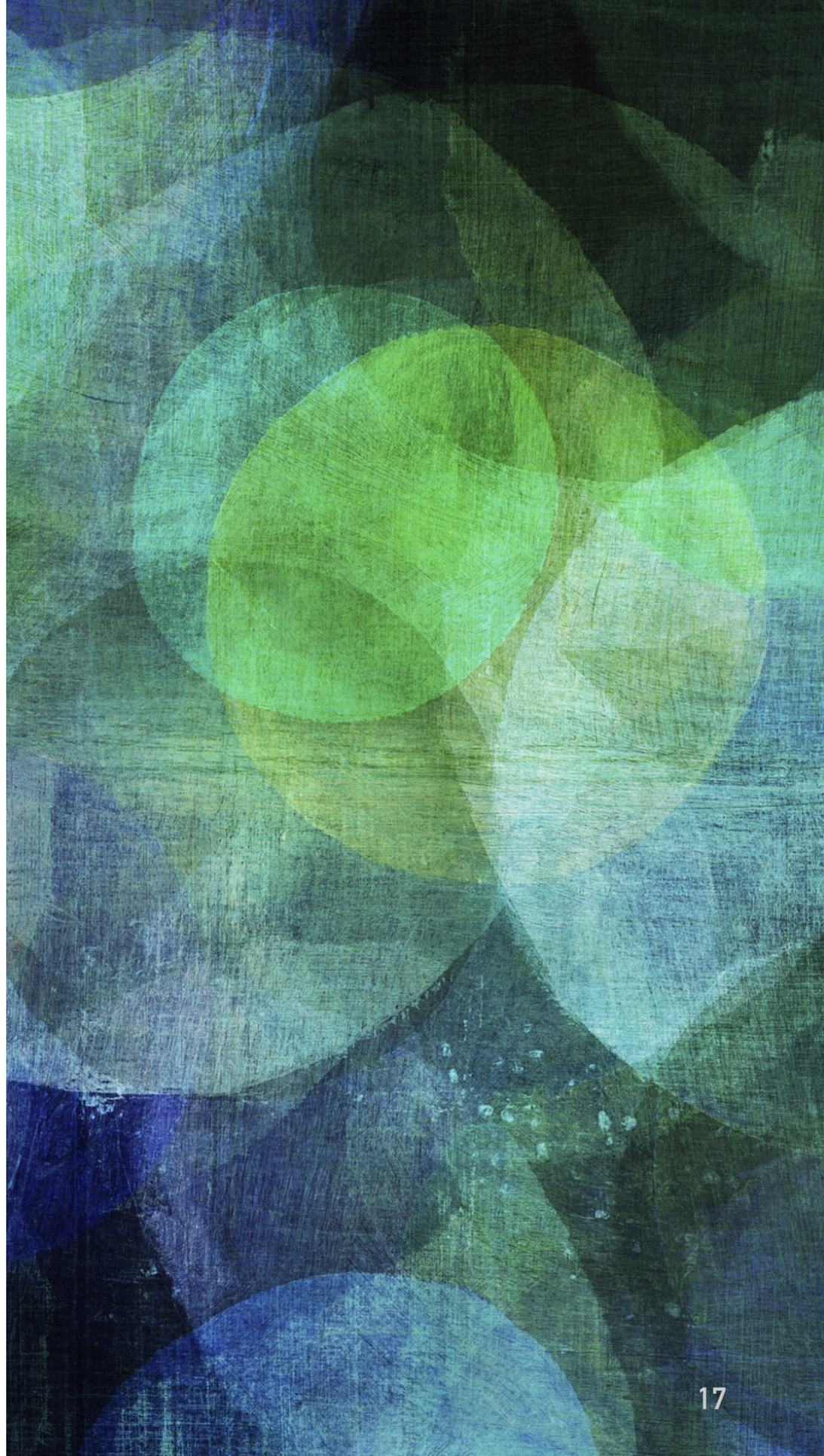


CONCLUSIONS AND NEXT STEPS

- DNN is performing very well, but it is strongly specialized on the issue it is trained on
- AE is also performing very well and it is promising for the detection of generic/unknown issues with a reasonable level of false positives
  - **NEXT: Test DNN and AE on further DT anomalies**
- Unsupervised learning is desirable because it can offer the chance to spot unforeseen problems, but it needs to be studies and understood better
  - **NEXT: Work on other algorithms, mainly unsupervised**
  - **NEXT: Extend the R&D project**
    - Extend the development to RPC in order to be able to check completely the TwinMux inputs
    - Extend the development to the TwinMux output
    - Cross the two informations and create a standalone monitoring tool able to determine the origin of the anomalies

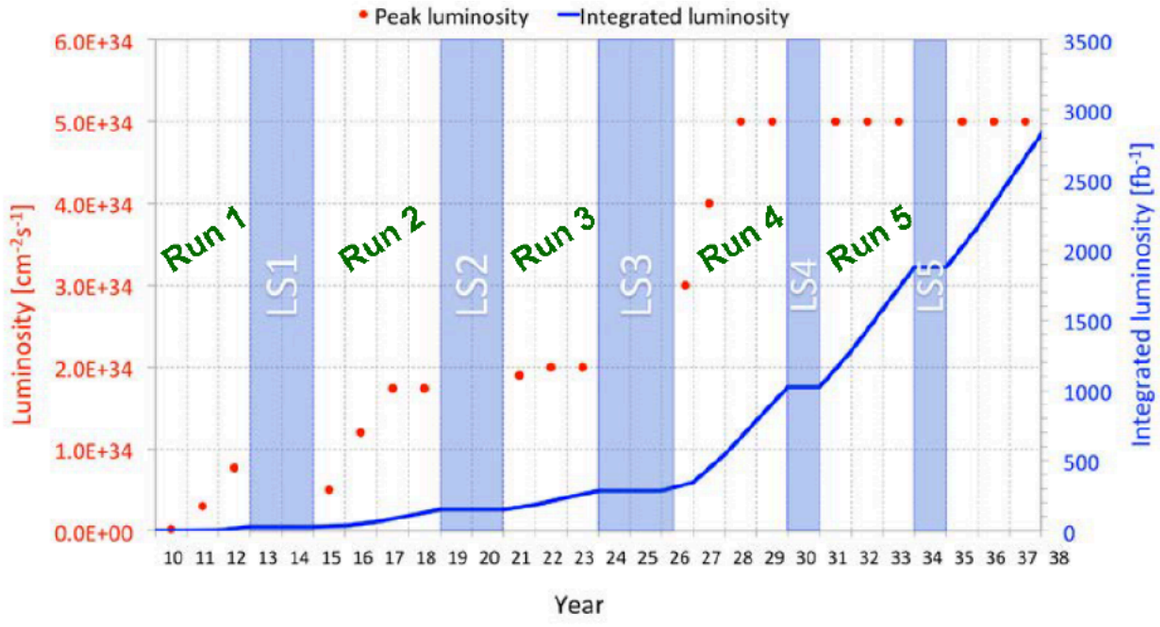
- **The python scripts to access the database and the jupiter-notebook used to train the model can be found here:**
  - **https://github.com/calabria/DTTriggerRateMonitoringWithML**

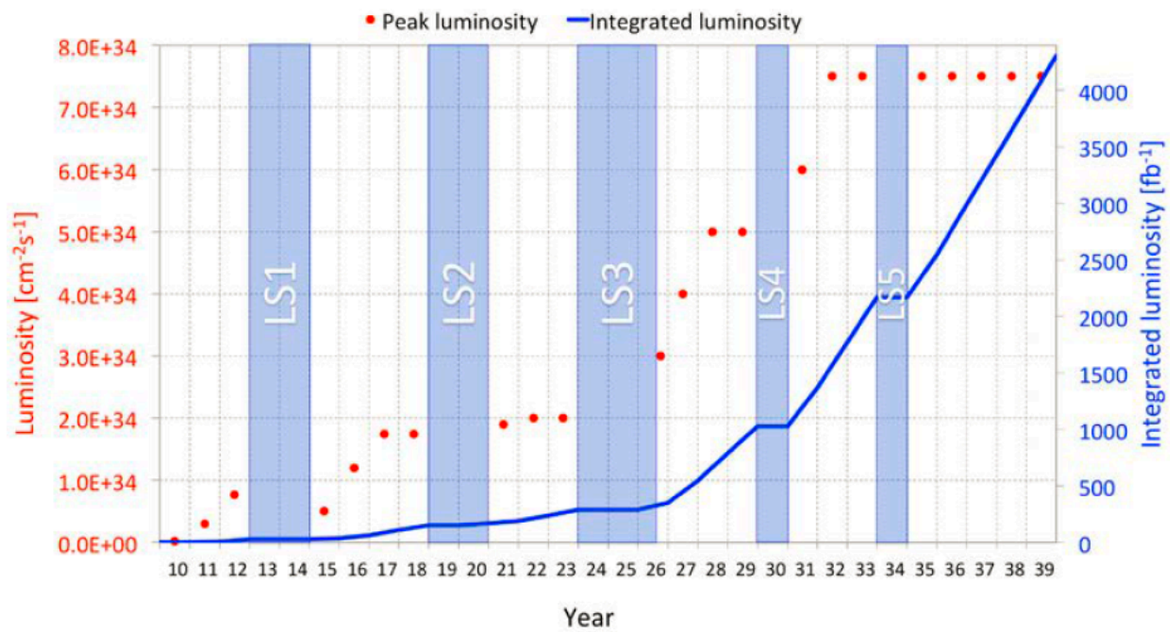
BACKUP



FUTURE CHALLENGES FOR HL-LHC

- **Higher instantaneous luminosity and consequently pile-up (up to 200 interactions per BX)**
  - Higher event size
  - More time needed for the pattern recognition algorithms
  - Increasing computing power both for the online selection and for the offline reconstruction
- **How we managed in the past:**
  - Increase computational and storage resources
  - There will not be financial resources to support this!
- **More improvements will have to be found in algorithm speed, by a combination of smarter algorithms and by making better use of parallel architectures, for instance:**
  - GPU accelerators and massive parallel programming
  - Machine learning algorithms
  - Performance tuning and software engineering

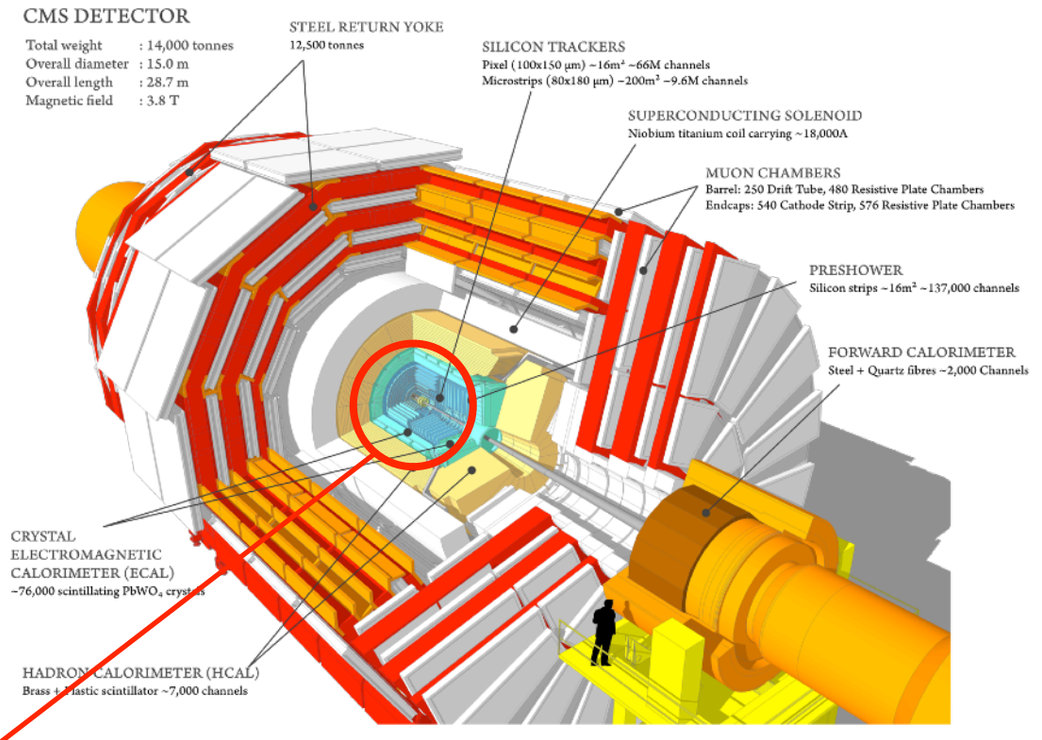


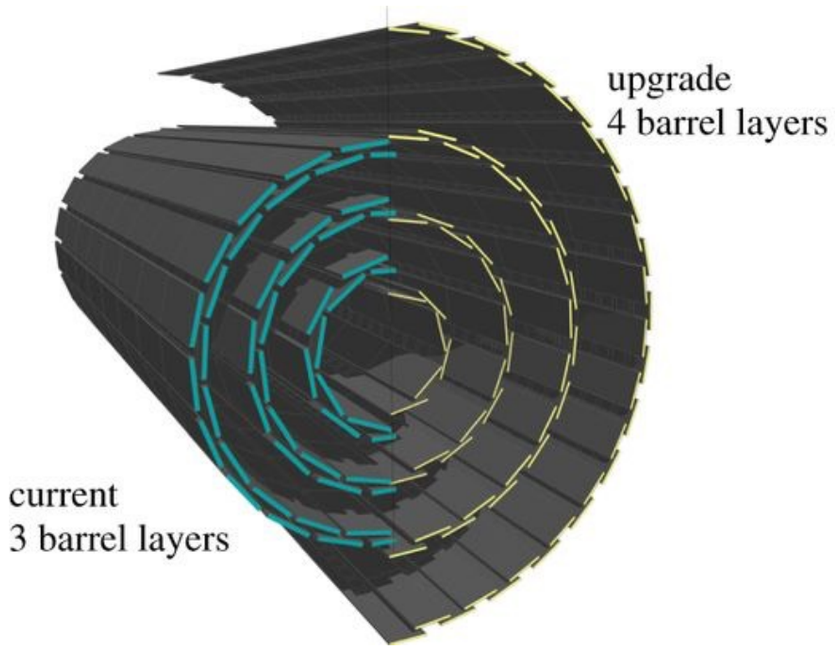


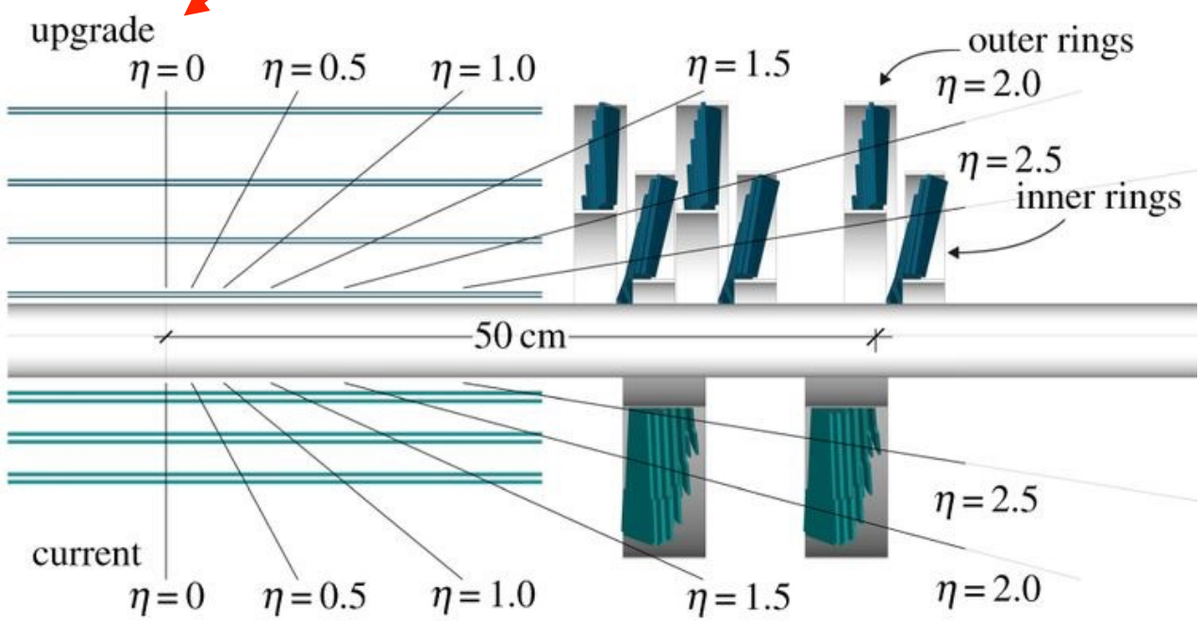
	LHC design	HL-LHC design	HL-LHC ultimate
peak luminosity (10^{34} cm^{-2}s^{-1})	1.0	5.0	7.5
integrated luminosity (fb^{-1})	300	3000	4000
number of pileup events	~30	~140	~200

THE CMS (PHASE-1) PIXEL DETECTOR

- n-on-n silicon sensor thickness: 300 µm
- Pixel size: 100 x 150 µm
- Four barrel layers instead of current three
- 3-disk forward system instead of current 2-disk
- Total Modules: 1856 (1184 + 672)
- Total Pixels: 124 million (79 M + 45 M)





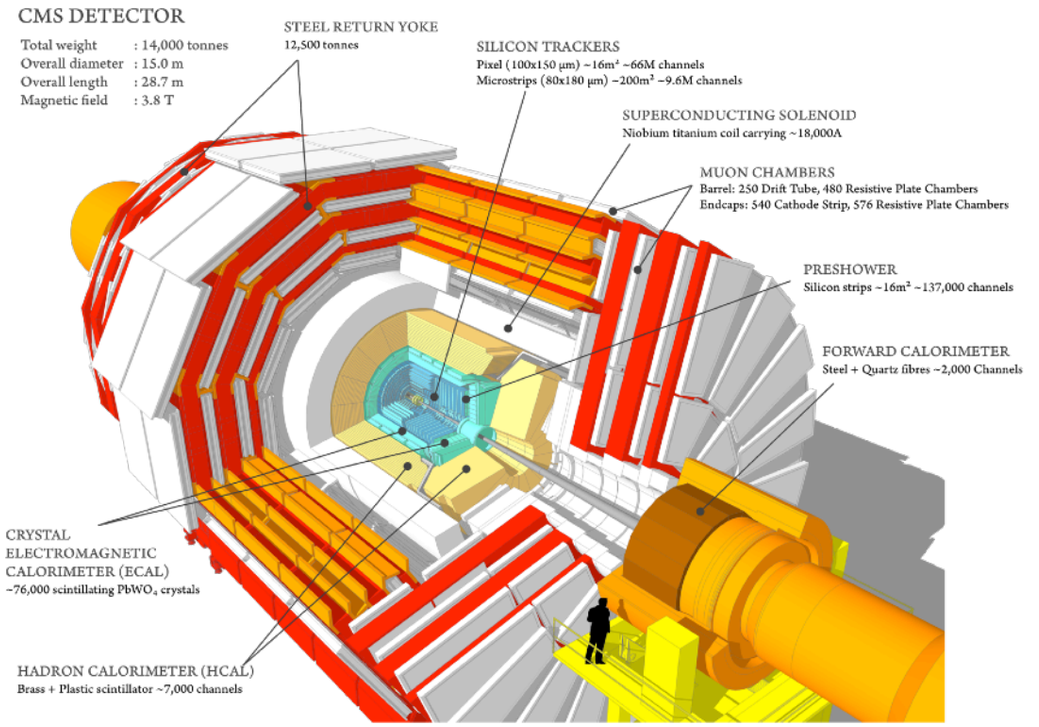


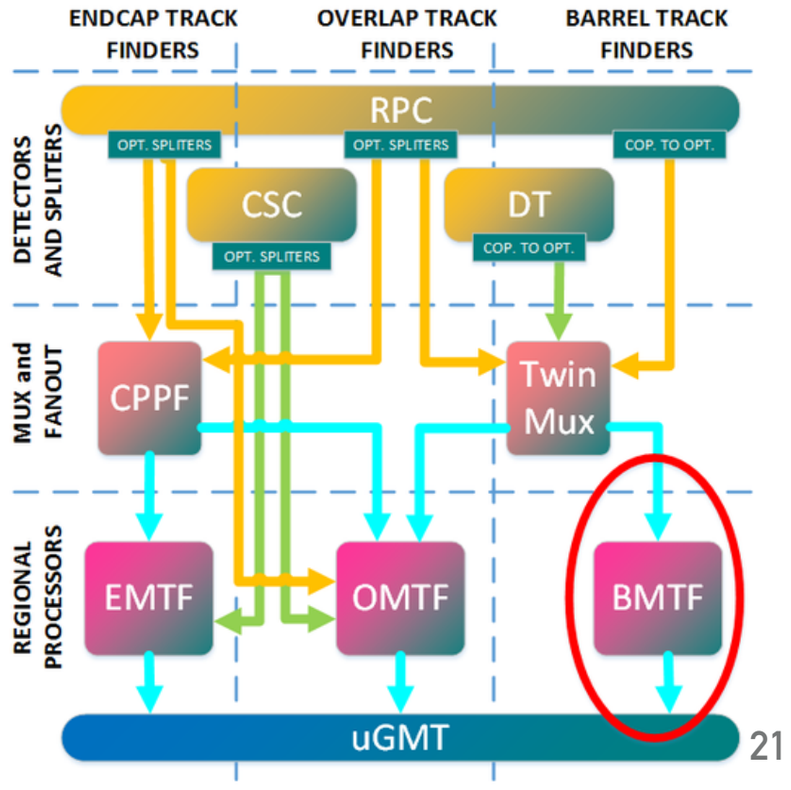
RAW2DIGI ON GPU: STATUS OF THE IMPLEMENTATION

- **Some details about the GPU implementation:**
  - **New GPU-friendly cabling map improves speed**
    - A GPU-friendly cabling map (basically a LUT) is generated and updated if it changes in the next event and copied again to the GPU memory
  - **Errors are treated and unpacked as in the serial code**
    - All the functions to check the status of the pixel rocs and recognize the type of error have been implemented as device functions (part of the kernel)
    - The possibility to exclude bad pixels and specific regions of the detector is also implemented
    - To this end the cabling map was also extended with the list of modules to unpack and the list of pixel bad rocs for the error unpacking
- **Optimized memory reserved for each kind of device array**
- **Optimized memory transfers packing the digi informations on the device and unpacking on the host avoiding to copy several arrays**
- **Using a GPU-friendly vector class instead of several arrays for the error unpacking**

THE CMS LEVEL-1 TRIGGER BARREL TRACK FINDER

- The muon barrel architecture groups the muon detectors in 12 wedges. Each wedge has five sectors and each sector, 4 DT detectors and 3 RPC
- The front-end electronics record muon primitives and send them to the TwinMux which concentrate data from different sectors. The TwinMux combines DT and RPC to create more reliable primitives which are called superprimitives. Then it fanout the data to the barrel and the overlap track finders
- The BMTF receive muon primitives from the DT and RPC detectors from the Barrel area of CMS ($|\eta|<1$)
  - The data primitives give muon coordinates, bending angle as well as quality bits that are used to evaluate the inputs
- The BMTF algorithm use the information to represent muon tracks and calculate physical parameters like the transverse momentum (pT), the total bending angle the quality of the track and the track addresses
- Each BMTF processor search for muon tracks in one wedge (own wedge) which may go also to the neighbor wedge (left and right)
- The algorithm runs in parallel for 2 muons in 6 sectors which correspond to 1 wedge (the sectors are 5 but the logic splits the middle to two). In the barrel there are 12 wedges. So it can find 2 x 6 = 12 muon tracks
- Every BMTF processor has a sorting logic which give the best 3 muons of the 12 possible tracks





A SIMPLE TEST CASE FOR DT TRIGGER RATE

- Testing some simple neural networks on a known issue:
  - Trigger board W+1, S4, MB3 is permanently off
- A supervised approach needs samples of normalies and anomalies for the training:
  - Take runs certified as “good” runs
  - Anomalies come for free in some sense, since the trigger board is always off
  - One can build the sample of normalies by exploiting the symmetry of the system and forcing the rate to the one of the symmetric trigger board, in this case: W-1, S3, MB3
    - [system, wheel, sector, station, rate, rate uncertainty, inst. lumi., lumi/rate, uncertainty on the ratio]

```
Normal chimney:
[2, -1, 3, 3, 4032.193, 53.42, 17987.064, 173.3344, 4.4609, 0.0731]
Anomalous chimney:
[2, 1, 4, 3, 3071.4035, 40.4563, 17987.064, 173.3344, 5.8563, 0.0956]
```

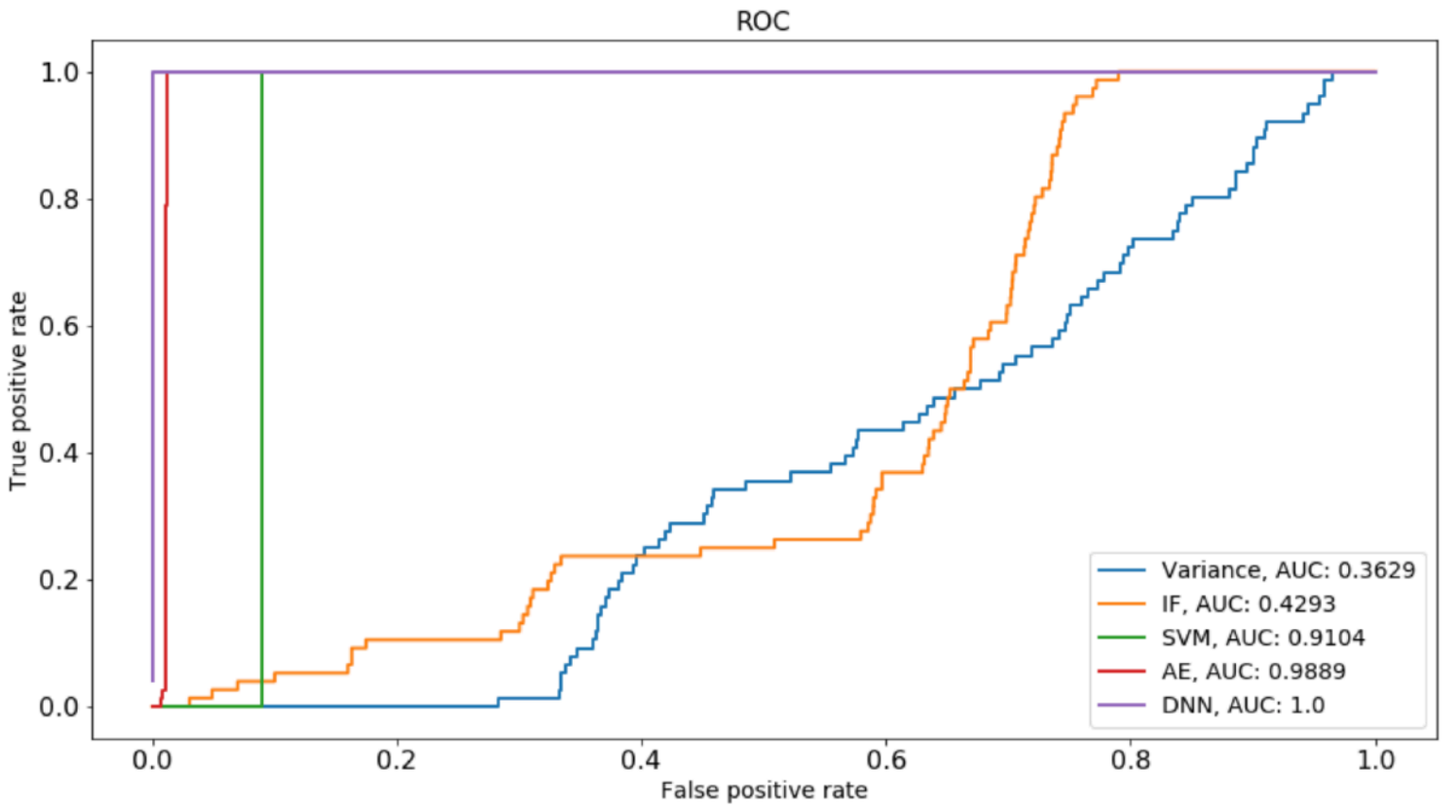
- Number of normalies and anomalies considered in this exercise
  - Very few faults, so anomalies and normalies are strongly unbalanced
    - Weighting properly anomalies and normalies
  - 20% of the data are reserved for the test

METRIC FOR PERFORMANCE EVALUATION

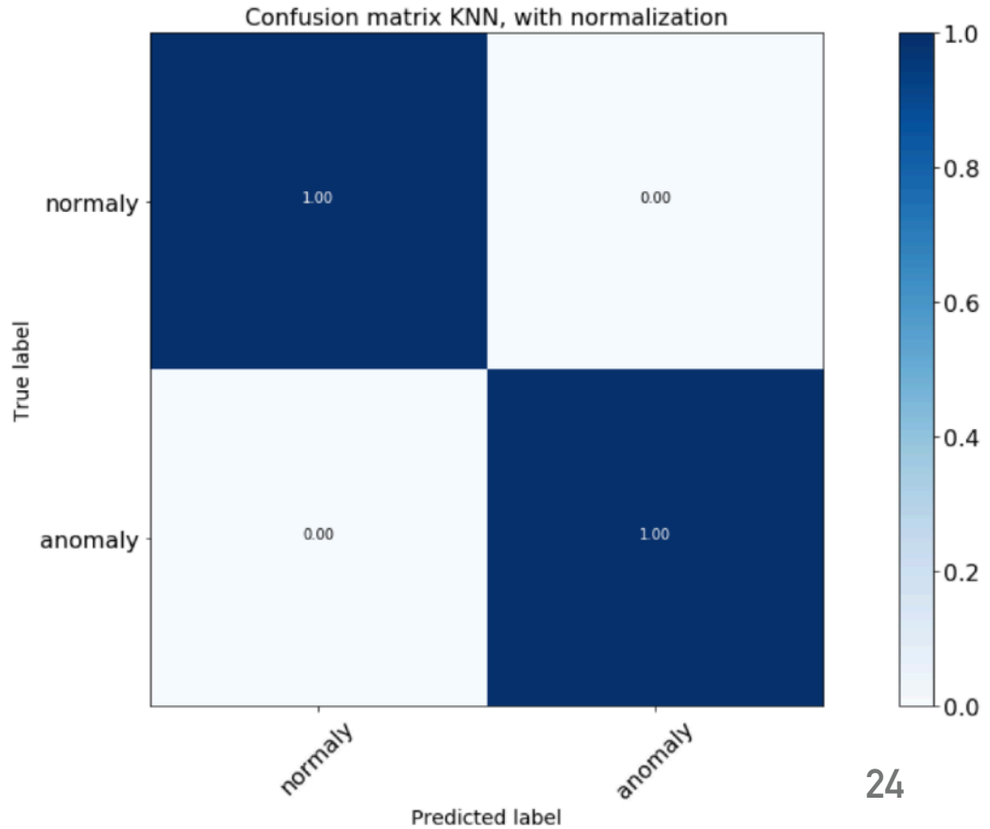
- **TP (true positive) is a correctly classified fault, while TN (true negative) is a correctly classified normal observation**
- Sensitivity TP/P: to keep high, i.e. maximize detection
- Specificity TN/N: to keep high, i.e. minimize false alarms
- Fall-out FP/N (1-Specificity): to keep low, i.e. minimize false alarms
- Receiver Operating Characteristic (ROC) curve and its Area Under Curve (AUC):
  - illustrates the performance of different classifiers when discrimination threshold is varied
- Deciding on the penalty of a false alarm versus false negative (or upper-bound false alarms) will be an essential in final implementation steps

COMPARISON WITH SOME BENCHMARK ALGORITHMS

- **Statistical: variance (probably not the best for facing this kind of problem)**
- **Outlier detection algorithms:**
  - Details : http://scikit-learn.org/stable/modules/outlier_detection.html
  - Classical machine learning: OneClassSVM (SVM) (need to perform a complete grid search for best parameters)
  - Unsupervised: Isolation Forest (IF)
- **Supervised nearest neighbor classifier (KNN): it performs very well but it is still a supervised approach**
  - Details: http://scikit-learn.org/stable/modules/neighbors.html#classification
- **DNN and AE still remains the best algorithms**



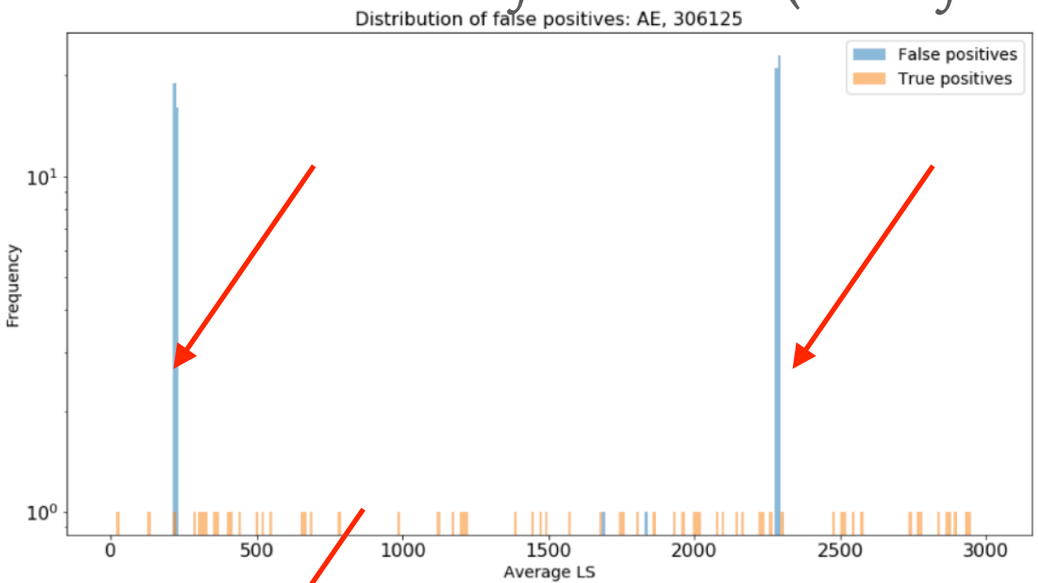

```
Normalized confusion matrix
[[ 1.  0.]
 [ 0.  1.]]
```

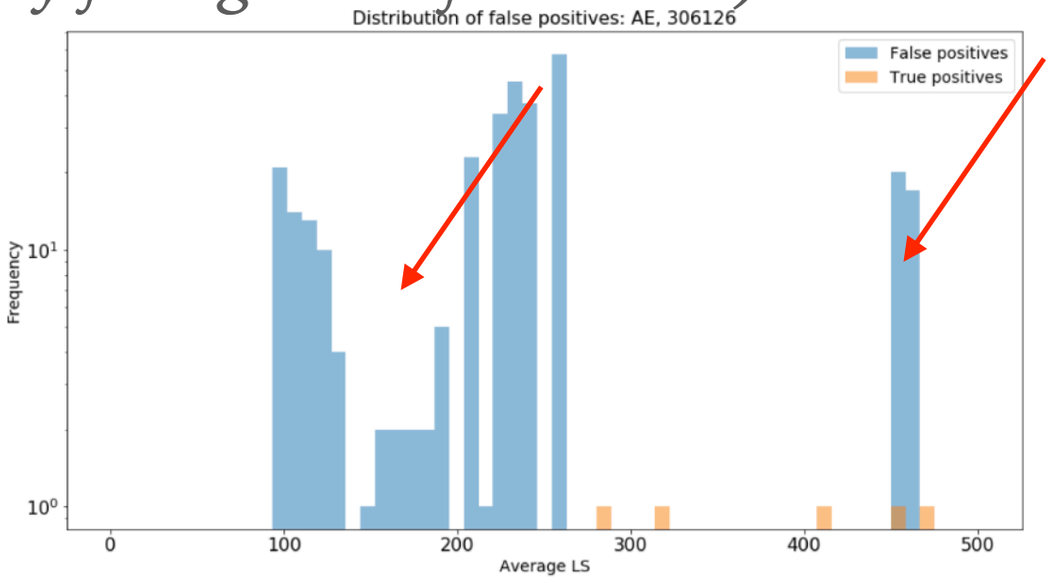


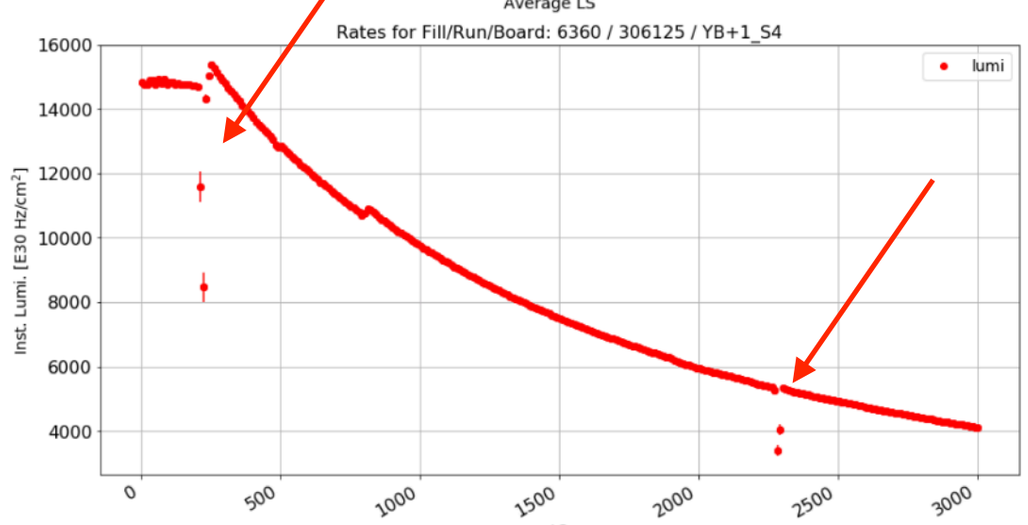
WHY IS THE AUTOENCODER APPROACH PROMISING?

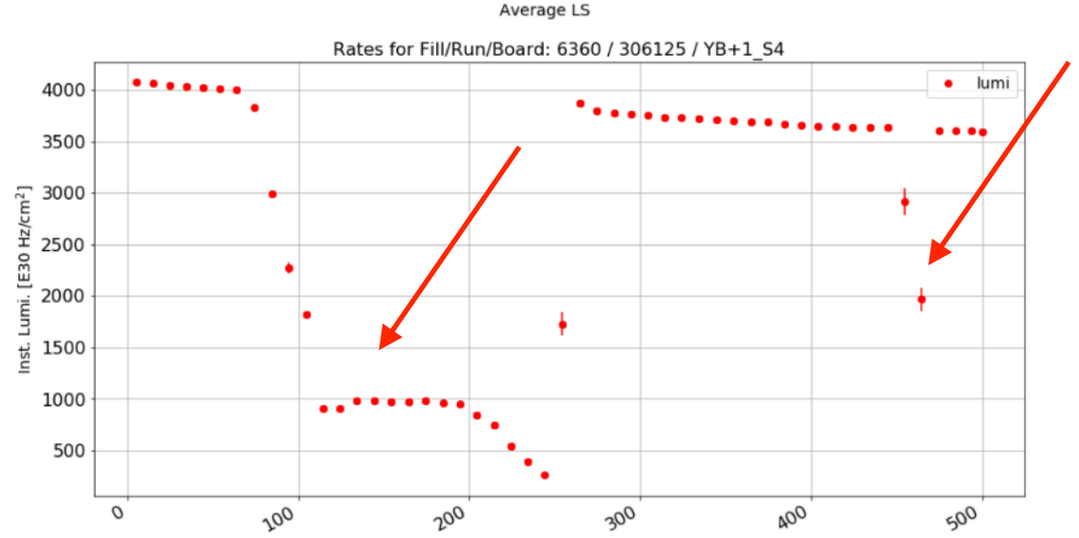
- DNN is performing very well, but it is trained against a specific issue
- AE is trained only using good data, so in principle is able to spot any kind of problem
  - AE is able to spot some luminosity oscillations during the fill, recognized as anomalies
  - This feature is not seen by DNN, since it is strongly specialized to find one type of issue

Distributions of the FPs (as defined by fixing a WP for the AE) vs. LS





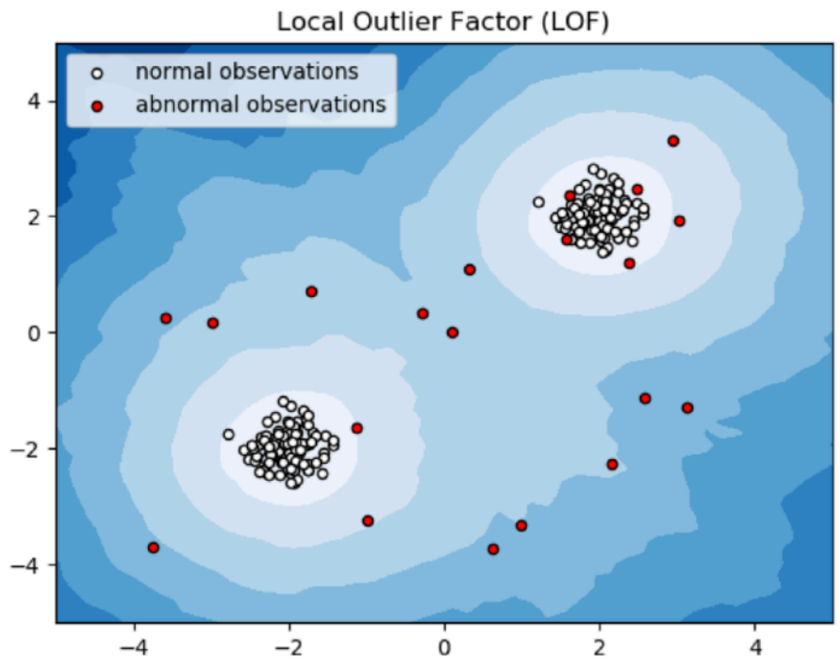


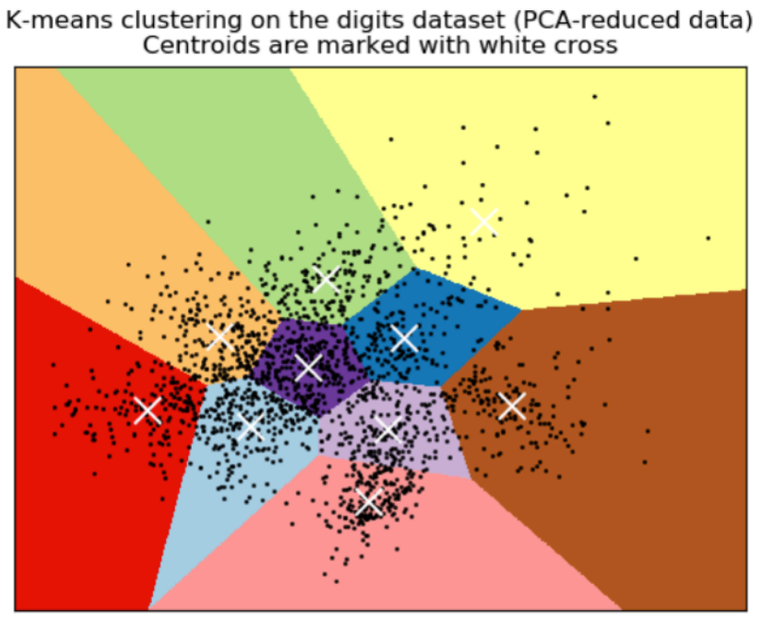


SOME DETAILS ABOUT THE UNSUPERVISED ALGORITHMS

- **The Local Outlier Factor (LOF) algorithm** computes a score (called local outlier factor) reflecting the local density deviation of a given data point with respect to its neighbors
- The idea is to detect the samples that have a substantially lower density than their neighbors
- The LOF score of an observation is equal to the ratio of the average local density of his k-nearest neighbors, and its own local density: a normal instance is expected to have a local density similar to that of its neighbors, while abnormal data are expected to have much smaller local density

- **The KMeans algorithm** clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares
- This algorithm requires the number of clusters to be specified
- The k-means algorithm divides a set of samples into disjoint clusters, each described by the mean of the samples in the cluster (called the cluster “centroids”)
- The k-means algorithm aims to choose centroids that minimize the inertia that can be recognized as a measure of how internally coherent clusters are





K. Androsov (Pisa): INFN Fellow presentation

flash talk

Past activities

Projects

- $HH \rightarrow bb\tau\tau$ analysis
- Pixel Phase 1 R&D and production
- Service work for tracking algorithms
- R&D: Tau L1 pixel trigger for Phase 2

Affiliations

- Dec. 2015 – Dec. 2017 INFN Fellowship for foreign students
- Dec. 2012 – Dec. 2015 PhD at the University of Siena

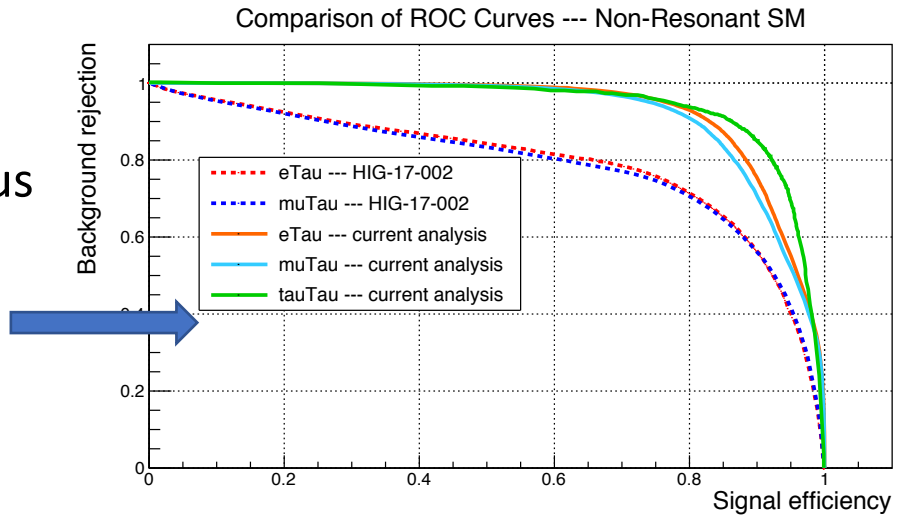
Current projects (Dec. 2017 - Now)

- Machine Learning in HEP:
  - Deep Tau ID and beyond
  - Data analysis
- Advanced pixel detector simulations for Phase 2 using GPU
- *Possible involvement in HPC tests at Cineca (under discussion)*

Machine Learning in HEP

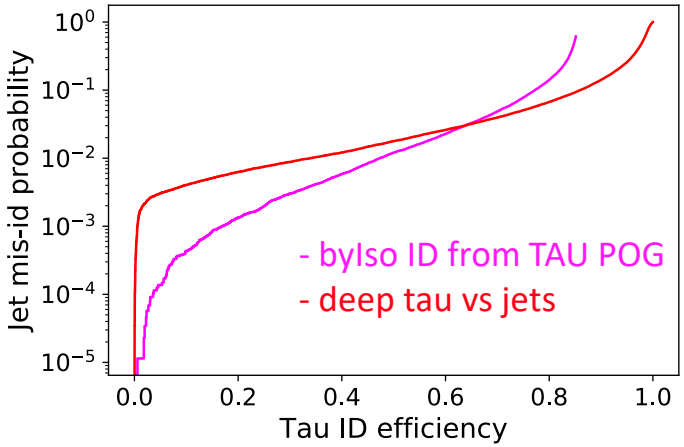
- What is the best way to apply Deep Learning in HEP?
- Can we afford “Zero Deep Learning” (without any human knowledge)?
  - This requires huge statistics, while full event simulations are CPU costly...

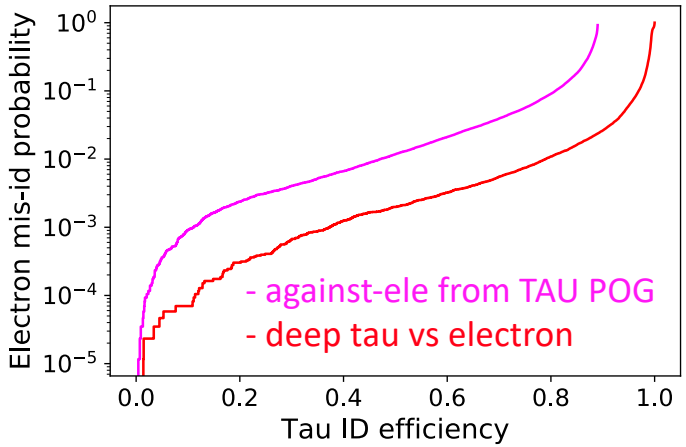
- How to pass our knowledge without adding significant bias?
  - Select relatively small set of discriminant variables based on mathematical algorithms from an extensive set of variables provided by “human experts”
  - Use those variables to pre-train inner layers of Deep NN or as an input of BDT
- We (me, A. Giraldi et al.) implemented an algorithm based on Jensen Shannon Divergence and Mutual Information measure that selects the most discriminating variables
- As a first try, this algorithm was applied to $HH \rightarrow bb\tau\tau$ analysis, keeping the same learner (BDT from TMVA) as in the previous version of the analysis:
  - Significant improvements wrt to the ”human expert” variable choice
  - The final expected limits for SM $HH \rightarrow bb\tau\tau$ improvement by almost a **factor 2**

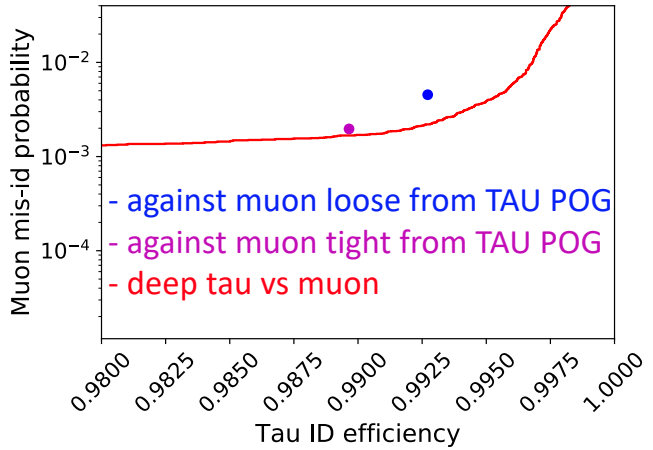


Deep Tau ID

- Physics objects reconstruction and identification is an excellent task for DL
- As the inputs, we can use a low level variables to not loose any information
- As the first target, the taus were chosen:
  - Current tau ID has 3 separate discriminators (against electron, muons and jets)
  - With DL I plan to introduce an unique multi-class discriminator
  - In the future, I plan to do full Deep Tau reconstruction
- To improve convergence (without introducing bias) we plan to pre-train the inner layers of the NN graph using algorithm described in the previous slide
- The full framework is in a very early stages of development, but first very preliminary results results of Deep Tau ID looks promising







Advanced pixel detector simulations for Phase 2 using GPU

- For Phase 2, the advances in the frontend design require sensors with smaller pixel cells and thinner active thickness
- R&D of such pixel detectors require detailed simulation to obtain reliable results. Within R&D we need to:
  - Find optimal pixel technologies and geometrical layouts
  - Test validity of the various radiation models
- 3D device modeling are very computational demanding
  - Licenses for simulation soft are very expensive and number of CPU is limited by 4 per license
  - On the other hand, simulation software allows to implement custom models as plugins
  - Most of the simulation algorithms are parallelizable => GPU isideal candidate to perform part of the calculation
- Within Pisa group, I'm starting to work on implementation of a plugin with GPU support for Sentaurus Device simulation TCAD.

Efficient and reliable data access using distributed and coordinated cache system

Sonia Taneja
INFN-CNAF

Personal info…

- Post-doctoral research fellow at INFN-CNAF, Bologna Italy
- User support - contact person for CMS
- Research and Development division (was part of INDIGO-DataCloud project)
- Started this fellowship - April 2018

Project and research interest

- Theme - Innovative Workflow and Data Management solutions for Large Scale science: large datasets, large workloads, heterogeneous platforms.
- Project- Distributed and coordinated cache system
  - Architecture - based on pool of distributed caches (provided by well connected WLCG sites), which are loosely coordinated by a central orchestrator to create an effective larger cache which will scale to better accommodate LHC needs for an efficient data access
  - Reduce latencies / Improve efficiency on remote data access
  - Reduced operational cost
- Present status:
  - Cache for http/WebDAV and StoRM (Nginx)
  - Collaborating with INFN-Perugia to converge on a generic cache solution

Future activities

- Automated deployment
- Exploring available cache technologies
- Customise the cache algorithms to match experiment requirements (Predictive analysis)
- To investigate AuthN/Z policies
- Implement federated cache
- Eventually test on commercial clouds