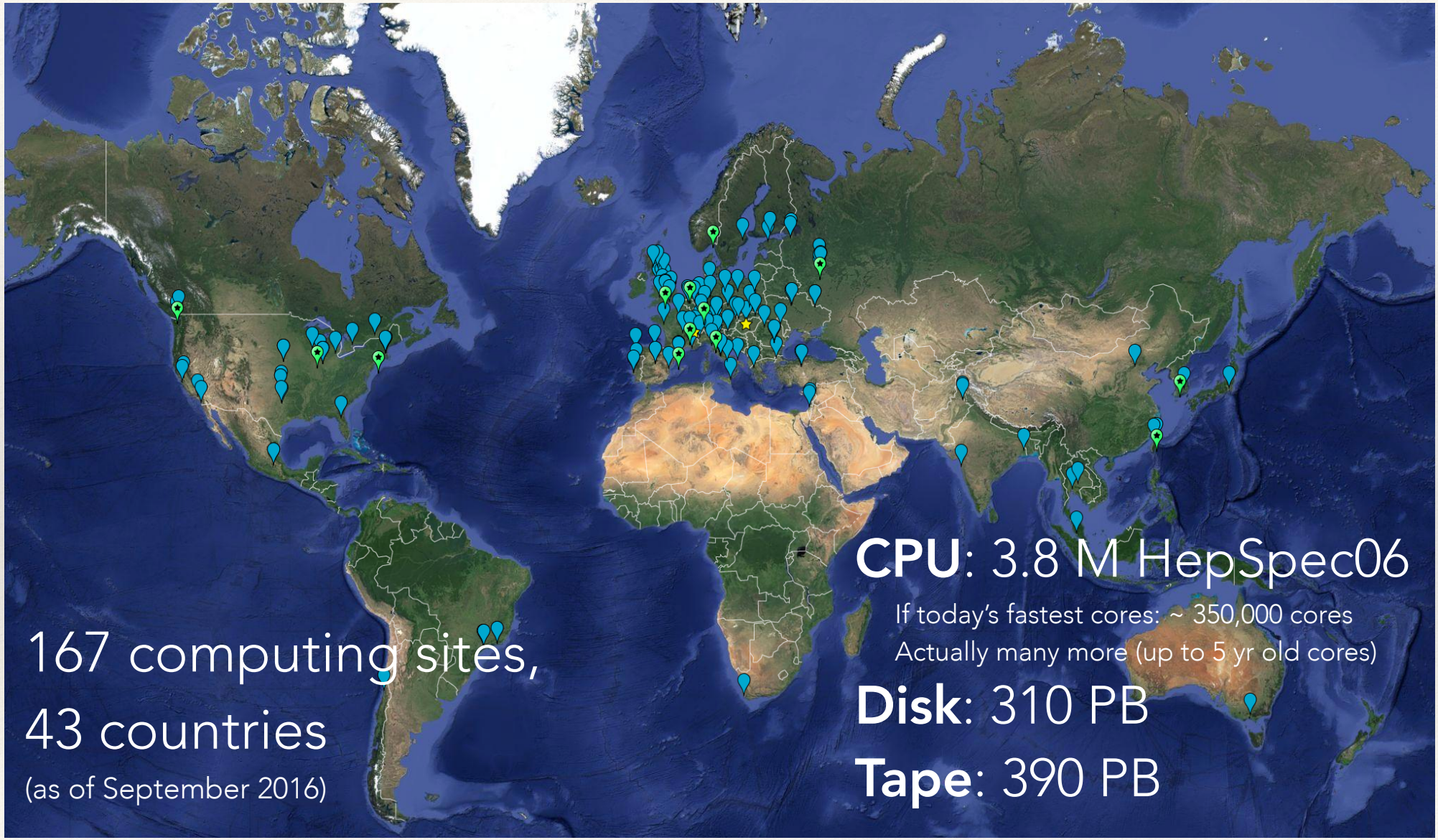
Computing and Technology Workshop - Vilnius - Dec 2016

Data Analytics in CMS

V. Kuznetsov (Cornell University, US)
D. Bonacorsi (Bologna University, Italy)

(.. credits also to students and other collaborators ..)

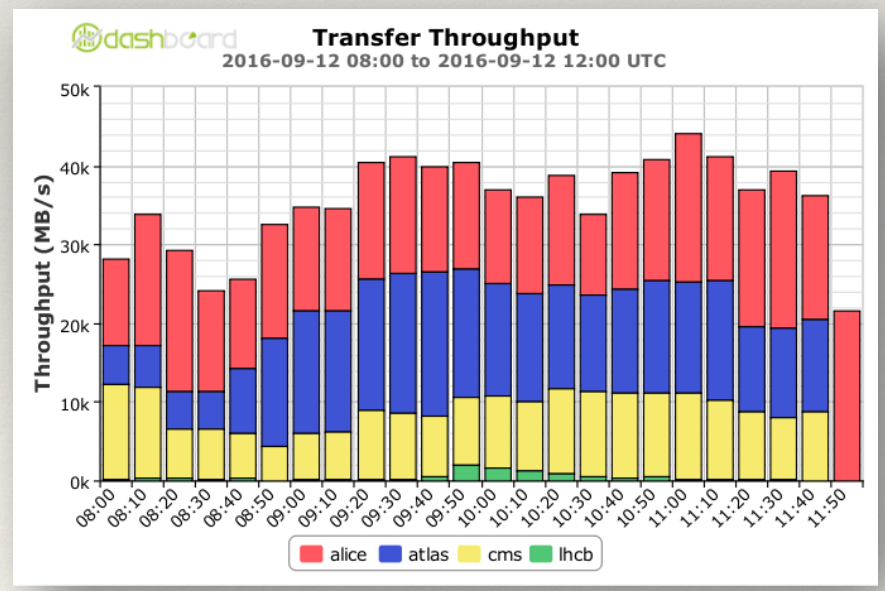
Worldwide LHC Computing Grid

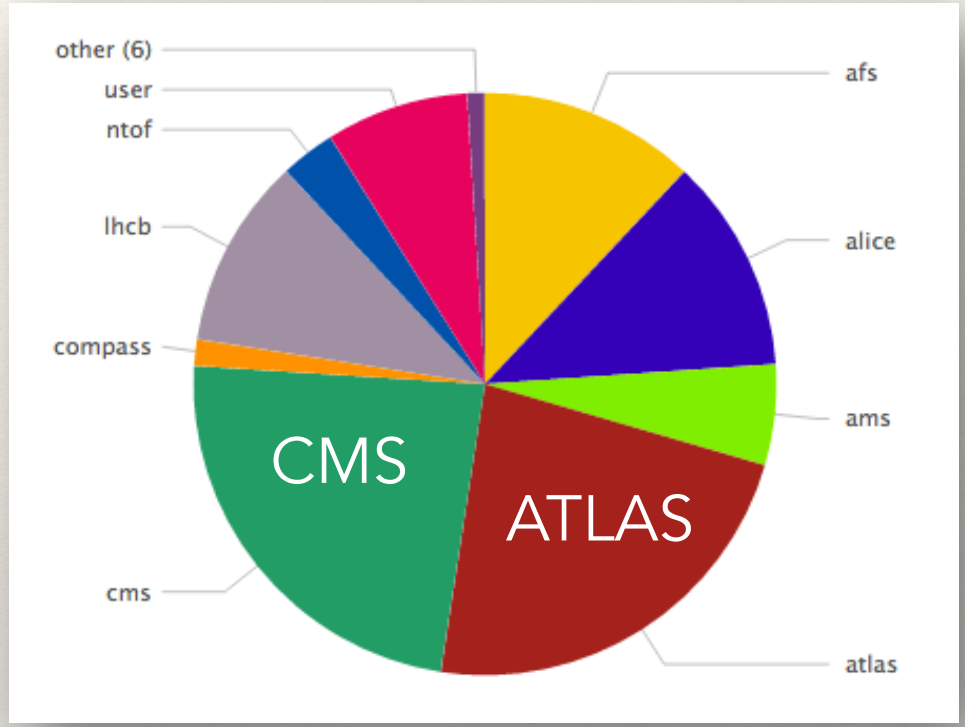


Big "numbers"..

Data taking continues to break records! Massive volumes of data written to storage

- June-August 2016 at **>500 TB/day**
- **10.7 PB** recorded only in July 2016
- 2016 to date: **~35 PB** LHC data
- CERN tape archive is **160 PB**





Data distributions over high-performance networks

- global data transfer rates increases to **>40 GB/s** (2x Run-1)
- regular transfers of **80 PB/month**
- many billions of files..

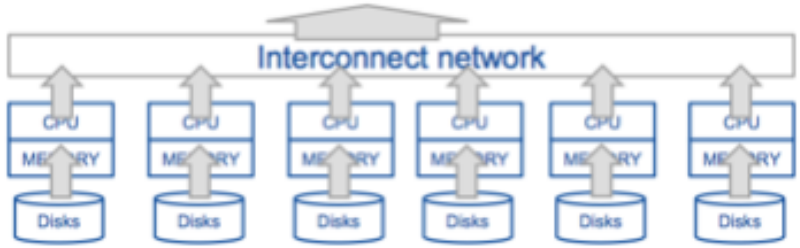
Hadoop

Apache Hadoop software library as an open-source framework for large-scale data processing

- distributed storage and distributed processing
- horizontal scaling
- optimization for high-throughput on sequential data access

Use cases:

- parallel processing of large amounts of data
- perform analytics on a large scale
- dealing with complex data: structured, semi-structured, unstructured



The project includes these modules:

Hadoop **Common**

- the common utilities that support the other Hadoop modules

Hadoop Distributed File System (**HDFS**)

- a distributed file system that provides high-throughput access to application data

Hadoop **YARN**

- a framework for job scheduling and cluster resource management

Hadoop **MapReduce**

- a YARN-based system for parallel processing of large data sets

Plus an entire ecosystem of other Hadoop-related projects at Apache, e.g.:

- Ambari, Avro, Cassandra, Chukwa, HBase, Hive, Mahout, Pig, Spark, Tez, ZooKeeper, ..

[credits: D. Bonacorsi, presentation on "Big Data in Big Science at CERN" to Accenture [Sep 2016]

Is CERN/LHC data "Big Data"?

The >30 PB/year of **physics data** are e.g not stored on Hadoop

- experiments data are mainly stored on tapes, with disks used mainly as caches

CERN and LHC experiments use Hadoop for storing the **metadata** of the accelerator and the physics experiments

Extract value from (meta)data:

- make (meta)data **accessible**
- draw **insight** from the (meta)data
- translate insight into **action**

CMS: towards **adaptive modelling**

CMS built a computing system that **worked** in LHC Run-1/2.

At which depth do we fully "**understand**" it?

- Can we perform precise modelling of specific workflows / site behaviours / system performances? Can we use this modelling to make predictions?
  - e.g. population vs pollution of Tier disks; TierX - Tier-Y data transfer patterns; ..

Computing operations (meta-)data is **all archived**

- but rarely (or never) accessed
  - e.g. transfers, job submissions, site performances, releases details, infrastructure and services behaviours, analysis accesses, ..
- we basically monitor to debug in near-time, not to analyse and learn from the past to design and build what's next

Here is where the Big Data zoology and **Analytics** techniques come in:

- a complementary "data scientist" approach

→ towards an **adaptive modelling of CMS workflows**

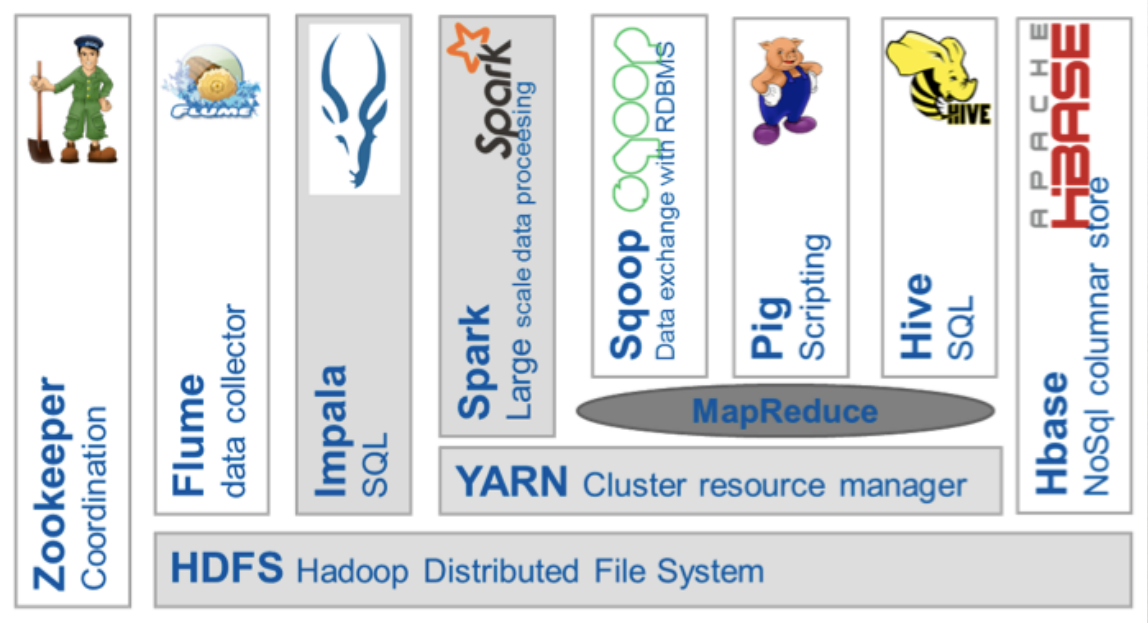
Many ideas came out of a first CMS R&D workshop in Bologna, back in June 2014

Hadoop at CERN-IT

.. and we also use the "Big Data" technologies installed at CERN for this.

Today experiments can use a **full Hadoop service** at CERN-IT

- set-up, operations, consultancy, user community
- full Hadoop ecosystem, most tools available, up-to-date

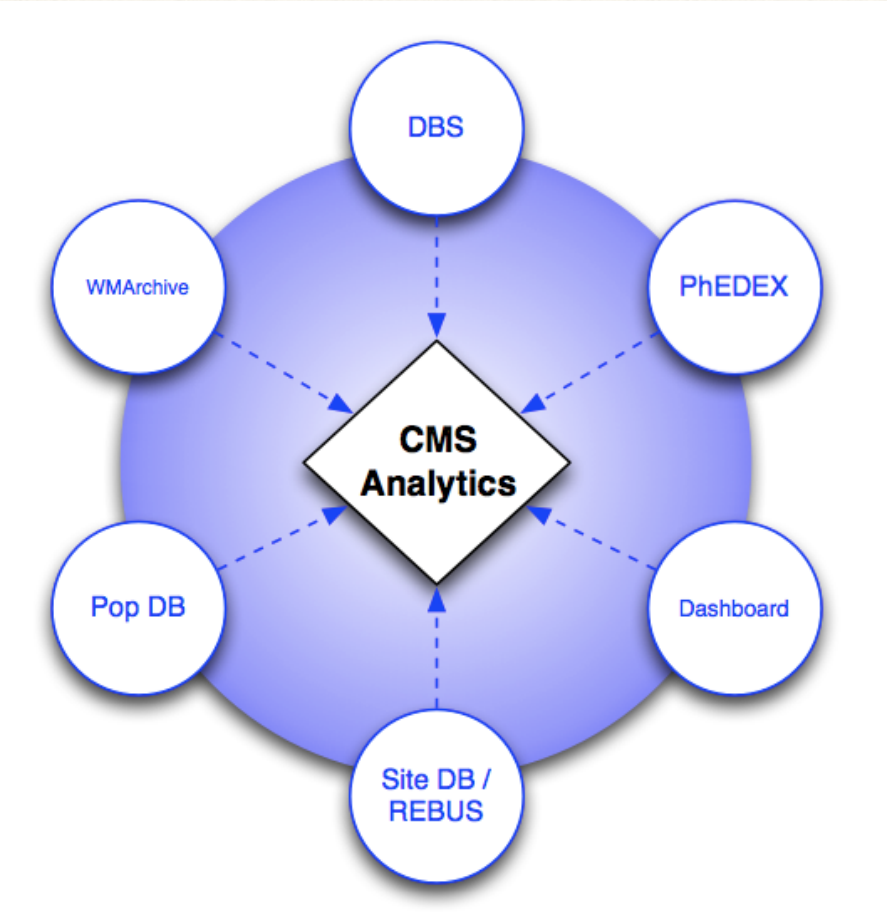


4 clusters in operations

- focus on addressing most pressing use-cases
- biggest is ~800 cores, ~1.5 TB mem, >2 PB storage

Very fluid and exciting situation, and new LHC use-cases pop up often!

CMS **Structured** data



Most ideas came out of a first CMS R&D workshop in Bologna, back in June 2014

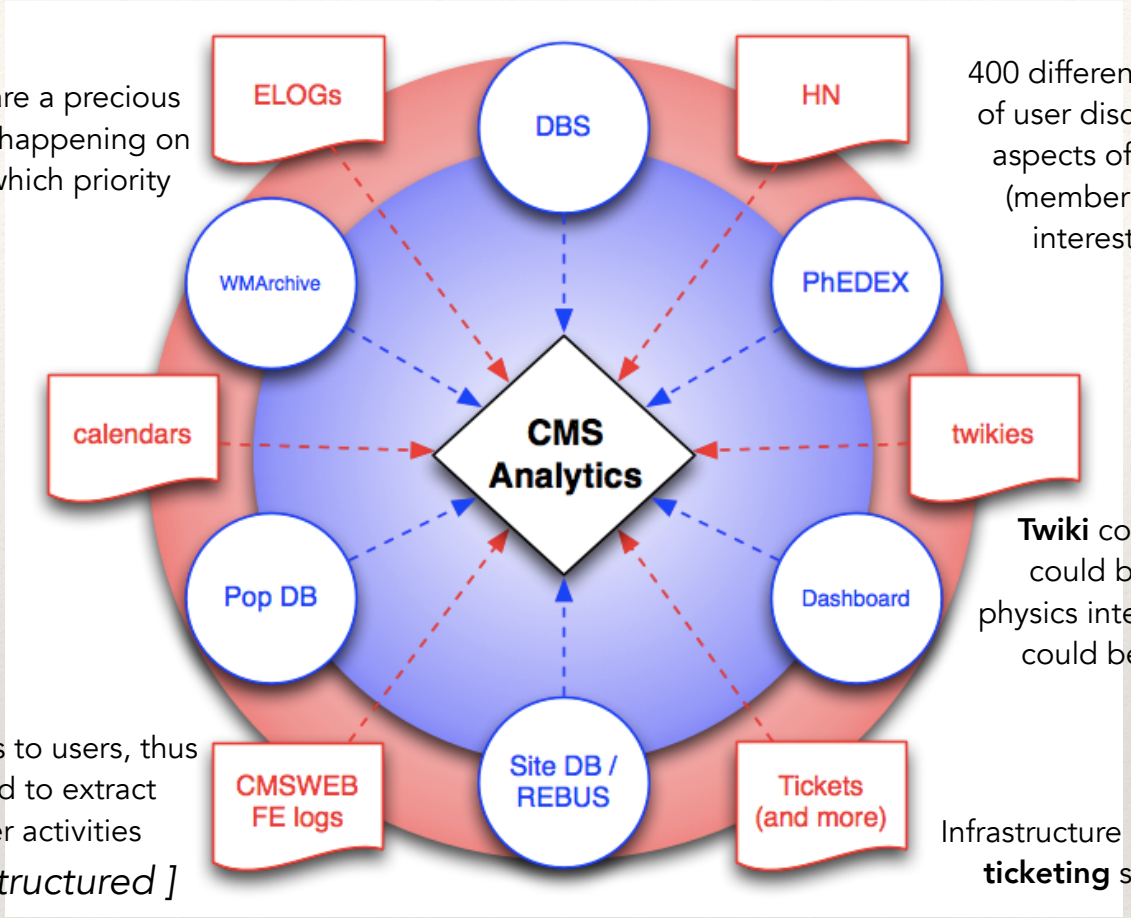
Structured info on a variety of CMS Computing activities are stored across multiple data services

- all info available via CMS data service APIs

CMS **Structured** and **Unstructured** data

Activity-based **ELOGs** are a precious source of info on what's happening on which systems and at which priority

400 different **HyperNews** fora, several yrs of user discussions, "social data-mining" aspects of collaboration-level research (membership changes study, physics interests evolution over time, …)



CMS events **calendar**, activity planning docs, list of major conferences and workshops, … could identify cycles within different physics communities

Twiki content as a knowledge graph that could be mapped to user activities and physics interests, and their evolution over time could be studied with appropriate tools.

It serves all data sources to users, thus its logs may be mined to extract valuable info on user activities

[Warning: semi-structured]

Infrastructure issues reporting/tracking, **ticketing** systems (JIRA, GGUS)..

Plenty of **unstructured** information in the CMS Computing (meta-)data ecosystem

- potentially very rich and sensitive predictors of user activities and future needs
- hard to process; manpower shortage; needs careful cost vs gain evaluation

Current focus is mostly on **structured**

Outline of next slides today

A (not exhaustive!) list of activities around Analytics, with focus on computing "structured" (meta)data on which CMS has use-cases and plans of work, and would welcome new collaborators

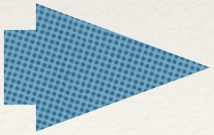
(NOTE: more activities exist e.g. in DeepLearning and BigData for physics use-cases, also in the CERN openlab context, etc - plenty of interesting work in progress!)

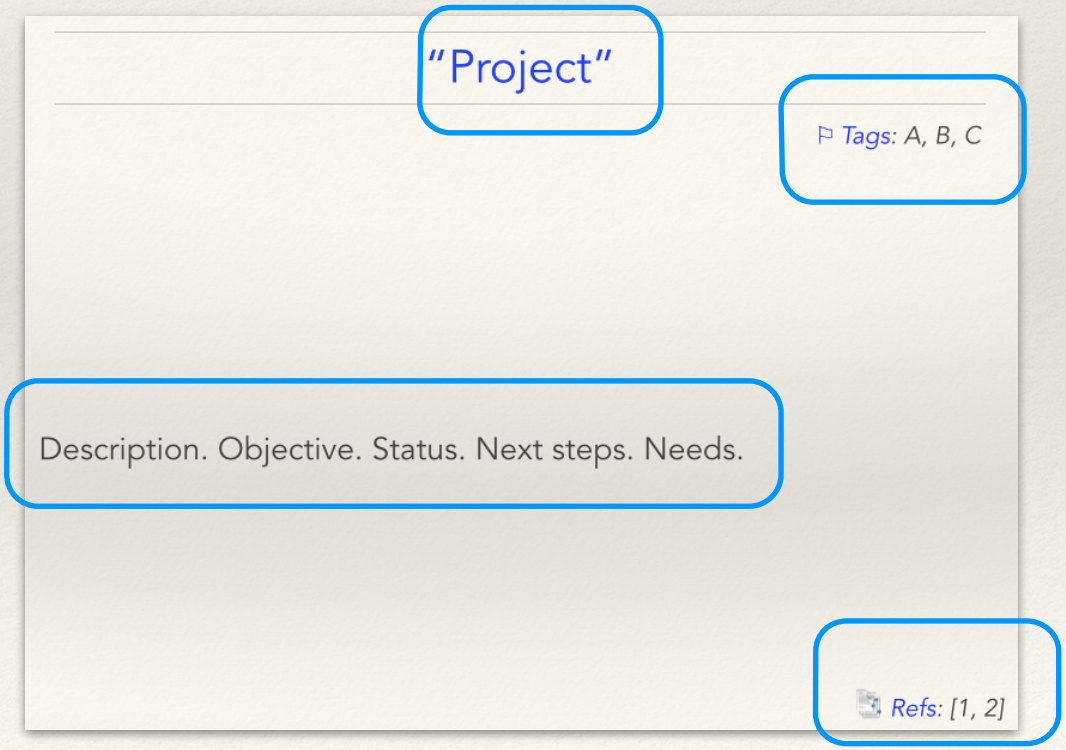
Goal today is to inform you and stimulate your potential interest

- if you have questions on specific area, contact us!

Template for each:

- see right-hand side





CMS dataset popularity with ML

⚐ *Tags: ML, SparkML, data-service, web development*

Objective: Study of CMS datasets access patterns

- clean disk-caches from unpopular data, trigger replication of popular data
- crucial to optimise storage usage on WLCG Tiers

(Valentin's) **DCAFPilot** [1] as the main ML-based tool (see next)

Supervised ML (classification) for the data popularity use-case

- define a 'predictable' popularity, seek for best ML algorithms, use ML to train a model to be able to predict popularity of current and future CMS datasets
- demonstrated that it can work - focussed work on MiniAOD format in progress

Next: We need to extend the work, plus to develop a data-service

- to be queried via API by end-users (e.g. Dynamic Data Management)

 *Refs: [1, 2]*

Engineering Effort for Effective ML

- From "Hidden Technical Debt in Machine Learning Systems", D. Sculley at al. (Google), paper at NIPS 2015

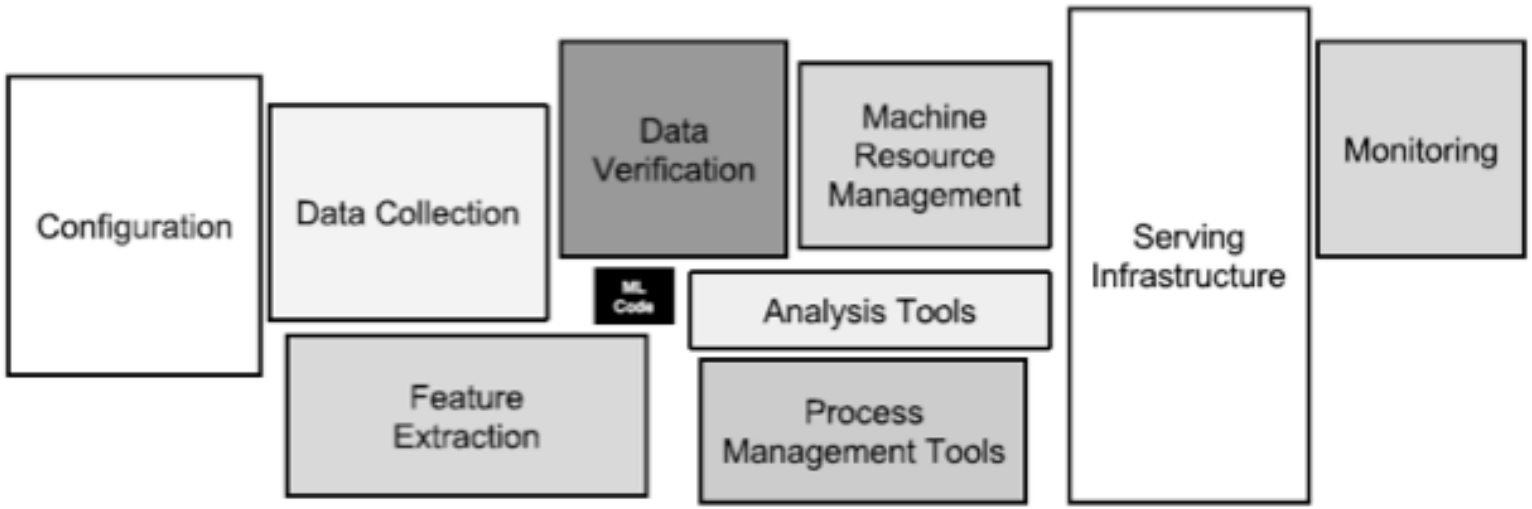


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

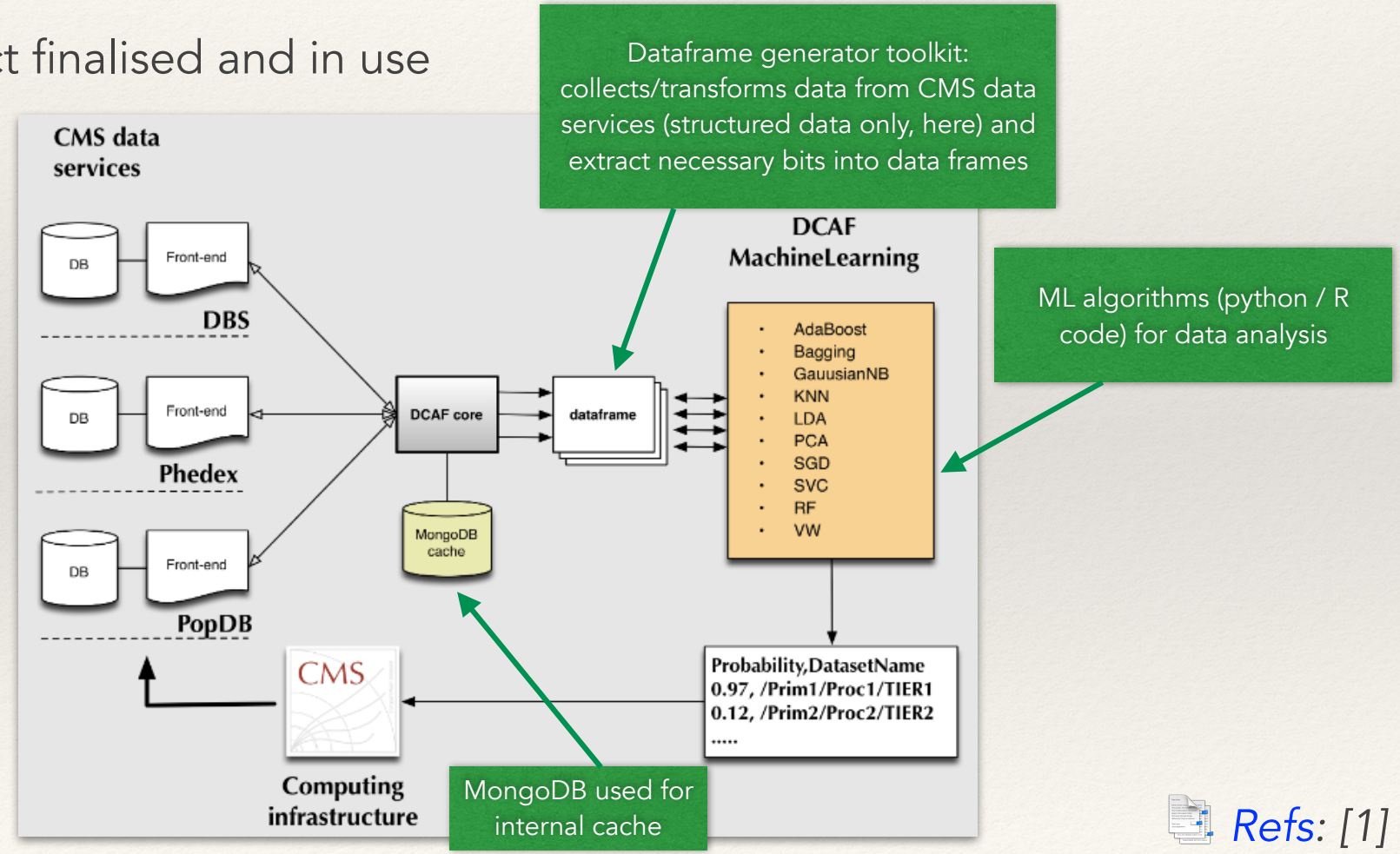
[credits: L. Canali, CERN-IT, Dec 2016]

DCAF

DCAF = Data and Computing Analytics Framework

Aim is to collect information from CMS data-services and represent it in a form suitable for "analysis" (e.g. ML tools)

- Pilot project finalised and in use



[credits: V. Kuznetsov]

Refs: [1]

DCAF (continued)

Framework consisting of several layers:

- **storage** layer, which can be used to keep information from various CMS data-services.
  - Currently, DCAF uses MongoDB and associated py-mongo driver as a storage solution, but it can be replaced with any other key-value store or RDBMS database
- **mapping** layer to provide mapping between sensitive information and its numerical representation
  - e.g. DNs into set of unique ids
- **aggregation** layer which collects and merge information from various CMS data-sources
- **representation** layer will take care of data representation suitable for analysis framework
  - e.g. represent our dataframe into CSV data-format
- **analysis** layer which will either use a custom analysis algorithm or provide bridge to other learning system libraries
  - The easiest solution would be to use python sklearn library [7] which already provides broad variety of learning algorithms

Refs: [1]

CMS data transfer latencies in PhEDEx

⚐ *Tags: ML, SparkML, data-service, web development*

Objective: Understand and classify issues behind data transfer tails

- based on **PhEDEx**, the current CMS dataset replication system
  - stateless and loosely coupled software agents, communicating with a transfer mgmt DB (Oracle backend)
  - PhEDEx handles subscriptions, performs (application level) routing, outsource transfer execution to gLite FTS, performs post-transfer checks and overall data placement monitoring and book-keeping
- large variety of PhEDEx nodes, data types, transfer volumes, priority windows..
  - "latencies" may originate in a variety of ways, actions are efficient only if based on actual nature of latency

Latencies **classifications** + **ML studies** of block replicas tails with DCAF

- extract FTS data, put into HDFS (.json raw format), convert into a table data format (.csv) to be read by DCAFPilot, which applies ML techniques

Next: proof-of-concept as a goal, then wrap-up the ML box into a data-service

 *Refs: [1, 3]*

CMS DBS/PhEDEx aggregations

⚐ *Tags: BigData, Hadoop, HDFS, Spark, data-vis*

DBS(v3) is the CMS event-data index ("*what data exists?*")

- a database and a user API, whose primary functionality is to allow for data discovery by CMS physicists, and to provide cataloging for prod+analysis ops

Objective: DBS/PhEDEx **aggregation** needs

- e.g. global CMS data volume stats, T1/2/3 data population and its evolution over time, recorded/produced evts counter per data types, data flow through sites (connected to the PhEDEx replica monitoring - see next), ..

Using CERN Hadoop+Spark platform

- quick progress recently! also cross-fertilising with neighbouring projects
- join soon if you are interested to learn quickly and help us to boost on this

 *Refs: [4, 5]*

PhEDEx dataset replica monitoring

⚐ *Tags: BigData, Hadoop, HDFS, Spark, data-vis*

CMS data recorded in files, organised in datasets and blocks

- **datasets** = set of files with common physics content (few files - 100k files; few GB - 100 TB)
- datasets are divided into groups of files called **blocks** to simplify DM (100-1000 files; 100 GB - 1 TB)

PhEDEx distributed the data over tens of Tiers, and tracks in Oracle the current status of every replica of every block produced in CMS

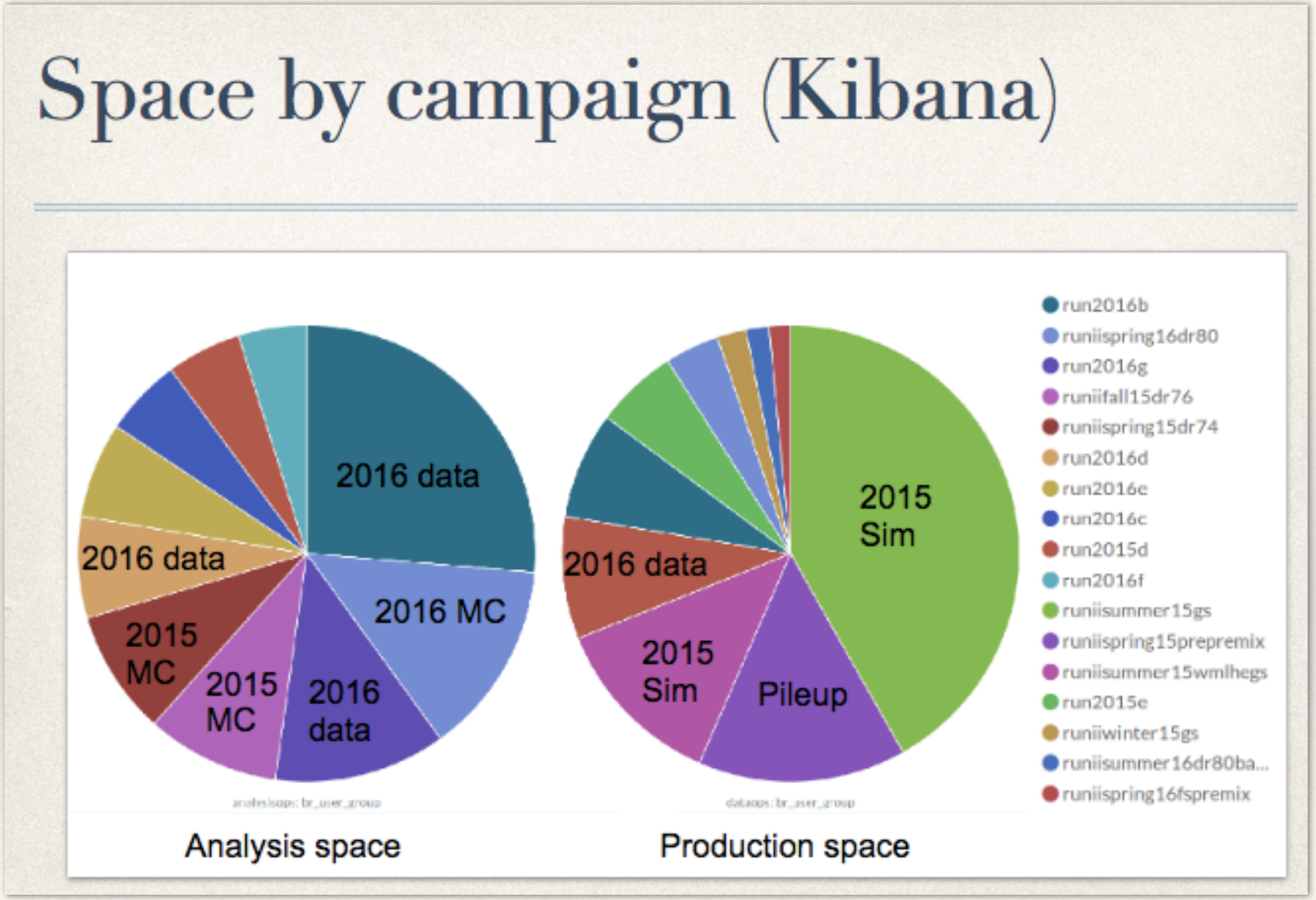
Objective: complement PhEDEx monitoring with a system to generate statistics about storage space occupation per data type and Tier

- export them to HDFS (CERN Analytics platform)
  - daily replica catalog snapshots on HDFS in CSV format; snapshot size: 2.0-3.5 GB, 5.5M-8.5M rows, 1yr => 1 TB
- Spark jobs for aggregation of block replica snapshots on HDFS at different levels/metrics

Next: set up a data-vis system to view both the current status and the time evolution of the results

 *Refs: [4, 6]*

PhEDEx dataset replica monitoring (continued)

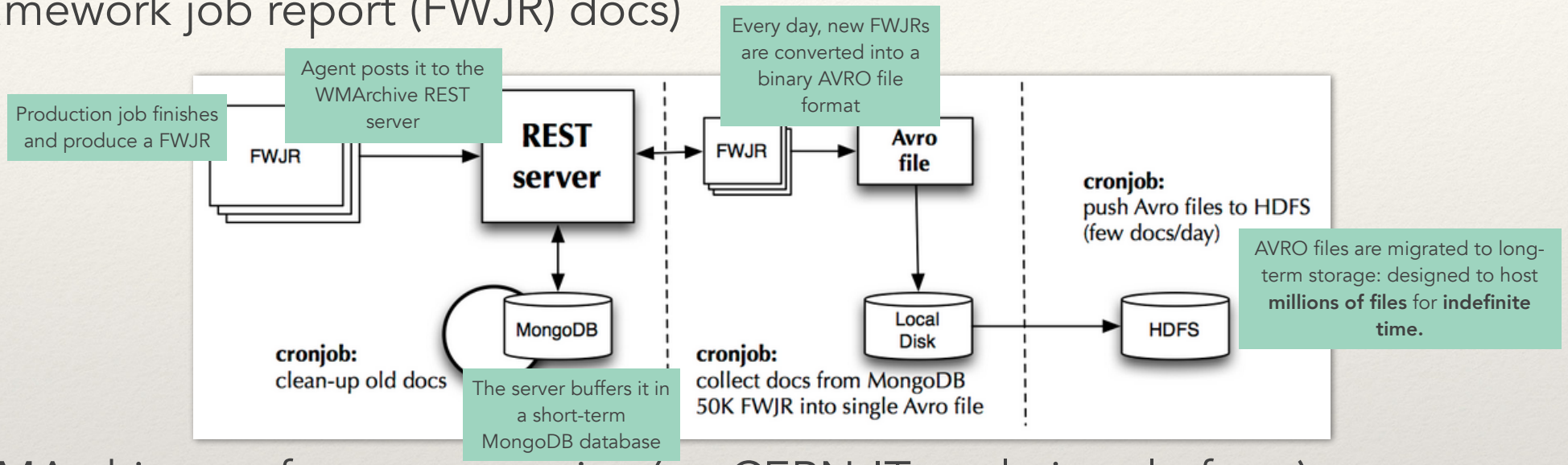


[credits: V. Kuznetsov, Sw/Comp week, Nov 2016]

WMArchive performance service

⚐ *Tags: BigData, Hadoop, HDFS, Spark, data-vis*

Part of the WMArchive project (aimed at a reliable store of CMS WM framework job report (FWJR) docs)



WMArchive performance service (on CERN-IT analytics platform):

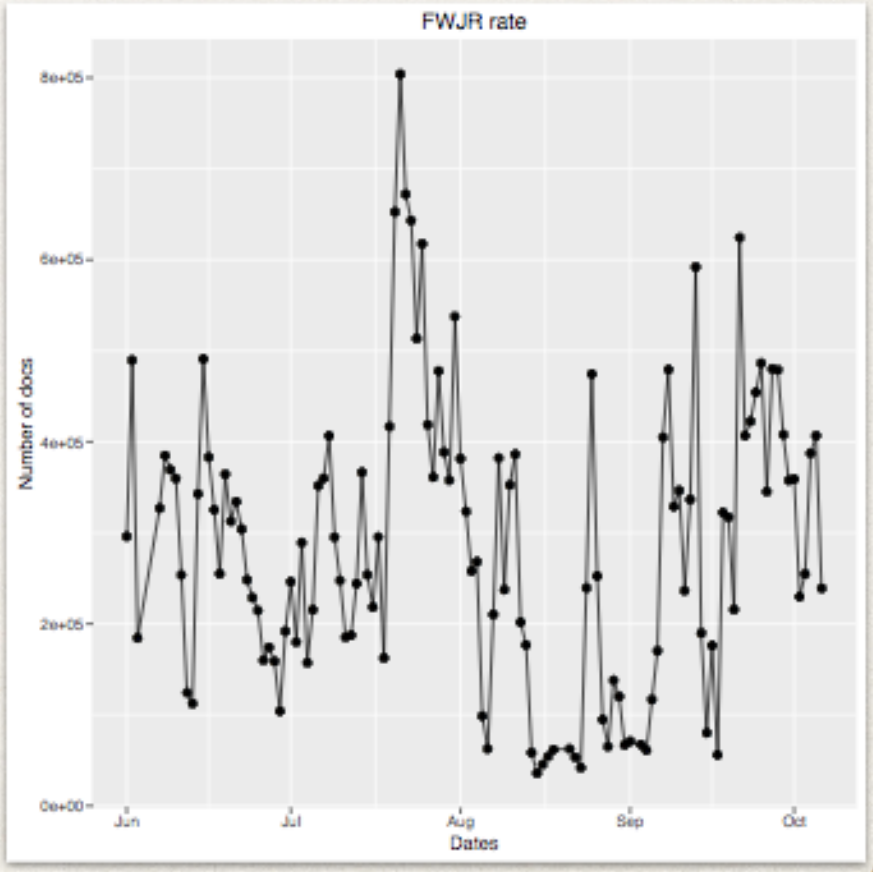
1. an aggregation pipeline to regularly retrieve data archived in HDFS, i.e. processes the distributed DB of FWJRs to collect performance metrics
   - clients interact with WMArchive via HTTP interface, post queries via JSON. Search and aggregation is done via Spark jobs
2. an interactive web interface to visualise the aggregated data and to provide flexible filters and options to assist the CMS data operators in assessing the job performances

Refs: [4,7,8]

WMArchive performance service (continued)

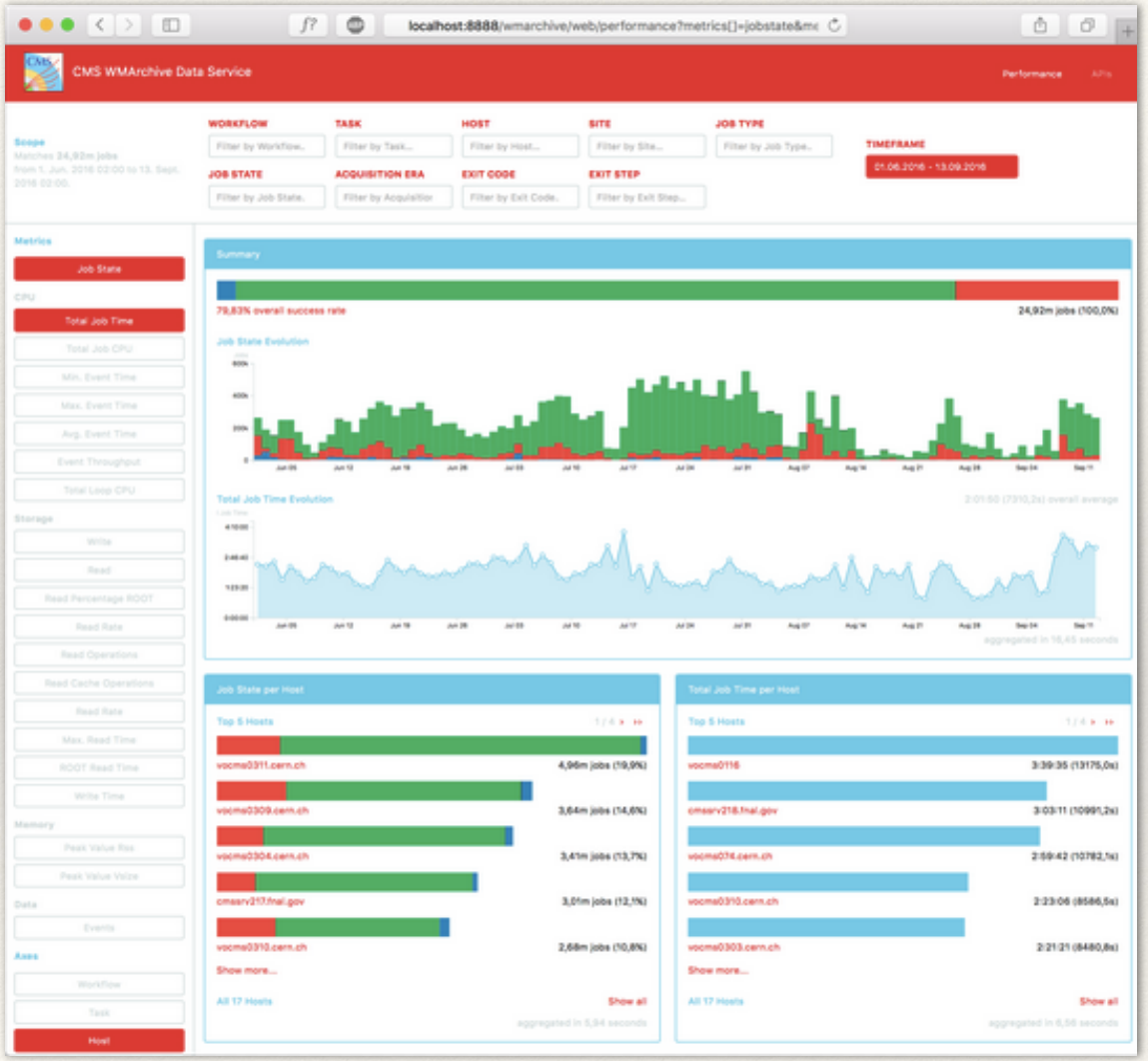
FWJR rate in WMArchive

- 7 production agents
- injection 24/7
- range from 35k-800k per day
- docs migrated from MongoDB to HDFS once a day
- 60k FWJR records per single (256MB) AVRO file
- Up-to-date we have ~50M docs on HDFS



[credits: V. Kuznetsov, Sw/Comp week, Nov 2016]

WMArchive performance service (continued)



[credits, Nils Leif Fischer (CERN Summer Student 2016)]

2

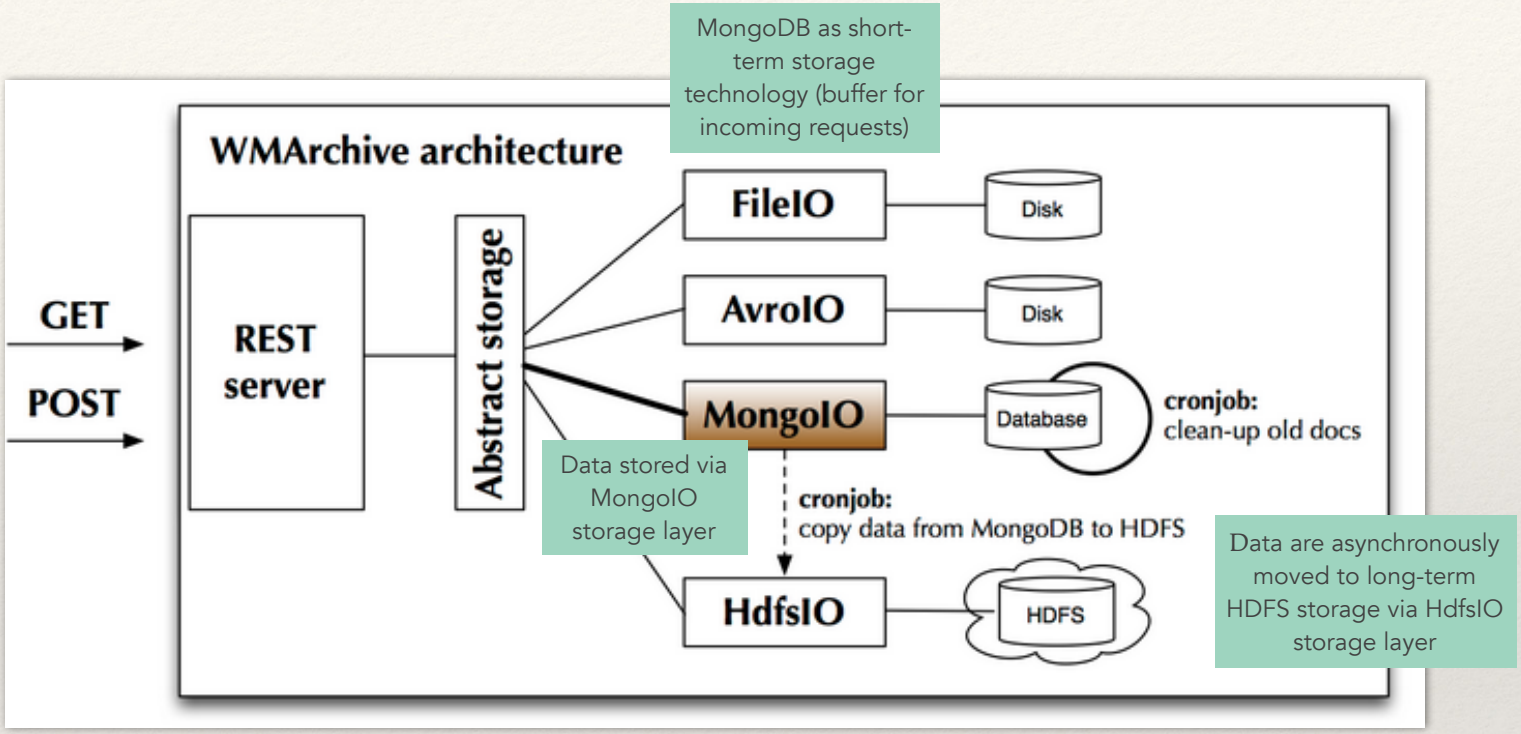
Technologies:

Big-data storage backend: Apache Hadoop
Aggregation pipeline: Apache Spark, MongoDB, Python
REST web server: WMArchive, Python
Frontend: JavaScript with Backbone.js, HTML, CSS, Sass
Visualization: D3.js

Snapshot of the web interface

Analytics on WMArchive data store

⚐ *Tags: BigData, Hadoop, HDFS, Spark, data-vis*



Objective: FWJR/DMWM/WMAgent analytics via WMArchive data store

- e.g. job failure evolution on sites, explore the dominant sources for job failures on a processing campaign, etc

Refs: [7]

CMSWeb Analytics

⚐ *Tags: data analytics, data-vis*

CMSWeb = centrally-operated cluster of web services critical for CMS Sw/Comp project operation

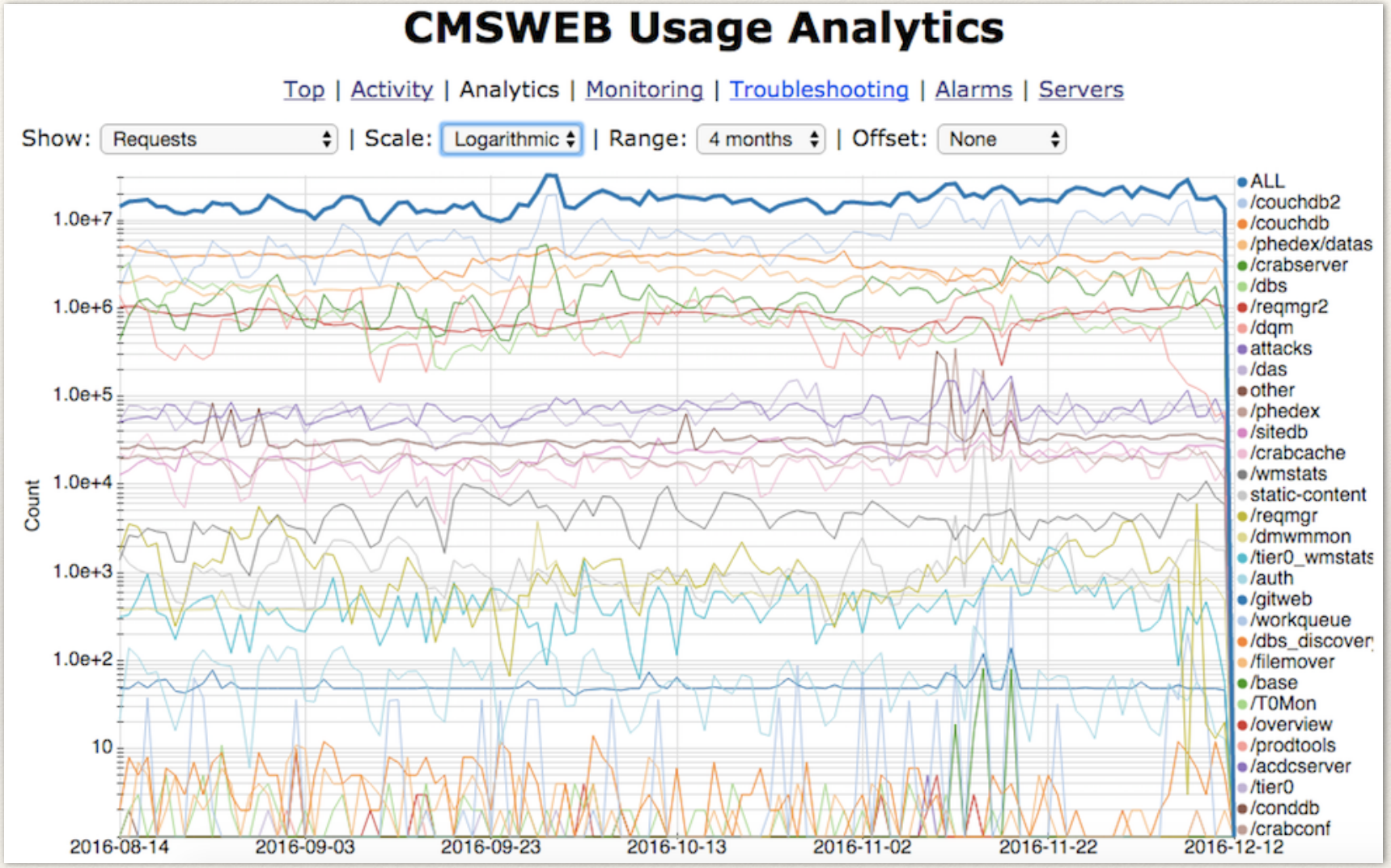
- DAS, PhEDEx, Request Manager, DQM components, Data popularity, SiteDB, and more..

Objective: CMSWeb analytics efforts

- we need to study CMSWeb logs to gain insight on user activities
  - primary goal: find cause of outages

 *Refs: [9]*

CMSWeb Analytics (continued)



[credits: V. Kuznetsov, Sw/Comp week, Nov 2016]

CMSWeb hosts ***O*(30) data-services**, with peaks of **100M requests**.

CMSWeb and concurrency

not strictly Analytics..

⚐ *Tags: asynchronous I/O, concurrency, web services*

Apart from analytics, there is interest in continuing a branch of work on CMSWeb:

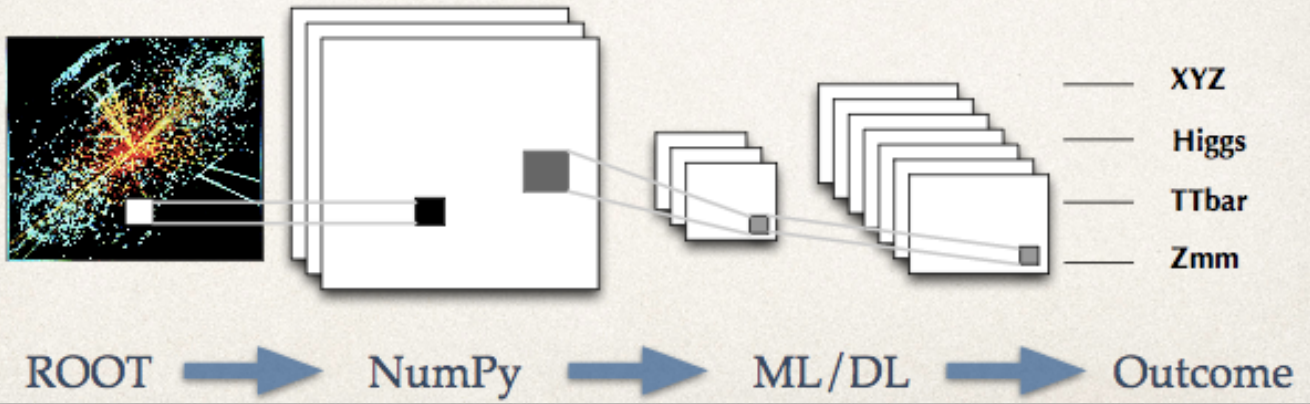
- generic Go sync/async web servers for CMS data-services
  - DBS/PhEDEx implementation in Go
  - performance studies of real data-services: Go vs Python
- Python async web servers for CMS data-services
- Go implementation of CMSWeb front-end (CMS auth stack)

 *Refs: [9]*

CMS event classification

⚐ *Tags: Deep Learning, algorithms*

- Machine Learning has been used in HEP for a while (ROOT+TMVA)
- Recent advances in hardware, methodology, software made big impact on adaptation ML in industry
- Deep Learning succeed in many fields, e.g. image, audio, speech recognition and classifications
- Outperform humans, find and learn patterns, easily applied across domains



[credits: V. Kuznetsov, Sw/Comp week, Nov 2016]

ROOT to NumPy transformation allows to use variety of ML/DL tools for real CMS data

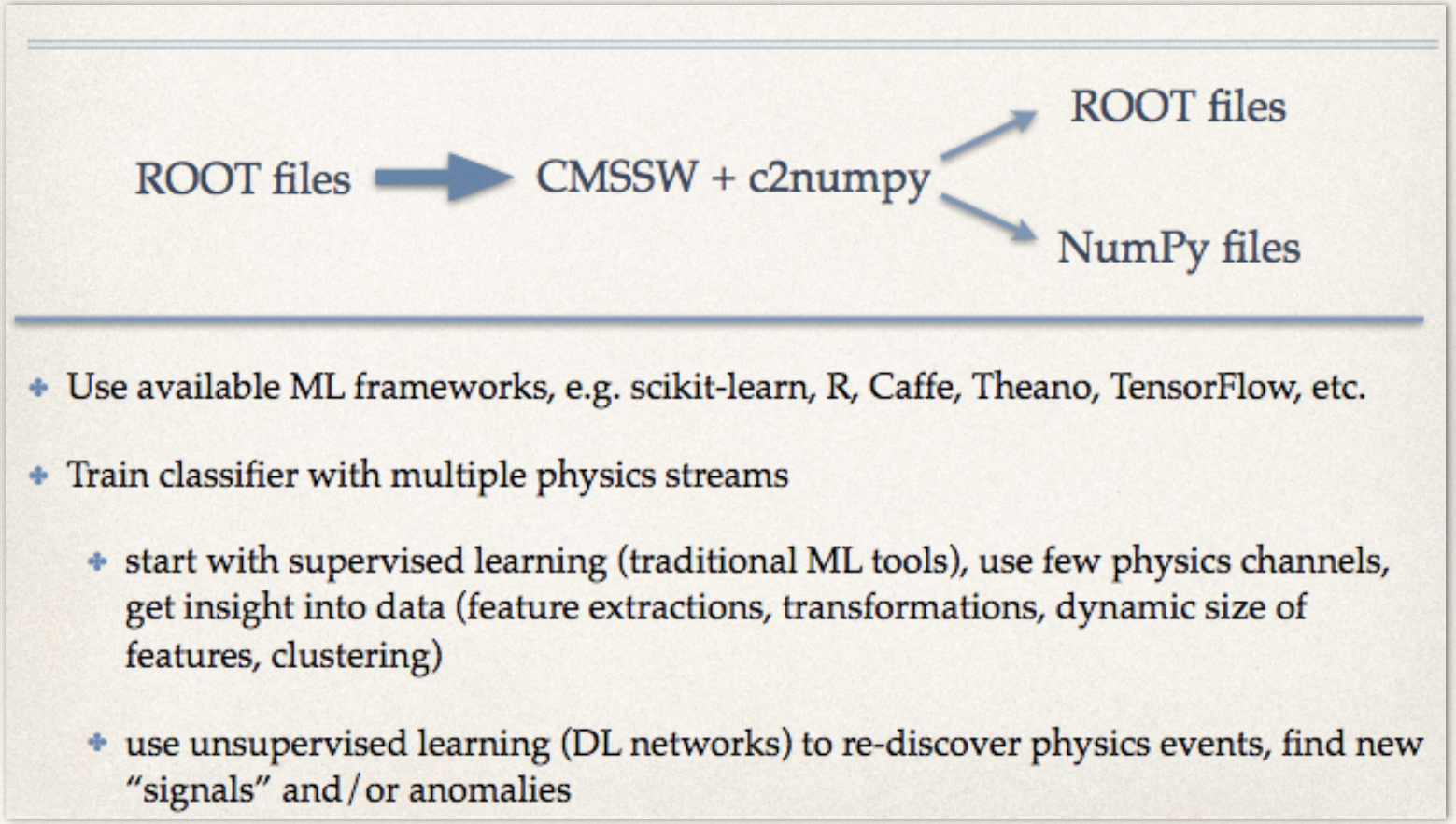
Caffe, Chainer, DL4J Deeplearning4j, KERAS, Microsoft CNTK, MatConvNet, MINERVA, mxnet, Purine, TensorFlow, theano, torch

Refs: [5,10]

CMS event classification (continued)

Objective: proof-of-concept that we can perform CMS event classification via Machine Learning / Deep Learning techniques by using CMS data (ROOT)

- recent project started at Cornell, connection with other similar efforts



[credits: V. Kuznetsov, Sw/Comp week, Nov 2016]

Refs: [5,10]

Conclusions

Data analytics in CMS has active work in progress on various fronts

- overall work plan progressing through well-confined and modular projects
- perfect fit for longer term partnership as well as students for few months
  - one aspect cross-fertilising the other
- very good support from the CERN-IT Analytics platform (growing quickly!)
- excellent environment to contribute with your data scientist skills, or to grow expertise on Big Data and an Analytics ecosystem
- stimulate a no-barriers fertile culture among **high-energy physics** (academia), **computer science** and **data science**

Contacts:

- daniele.bonacorsi@unibo.it, vkuznet@gmail.com

Not exhaustive at all..

In the **back-up**: some more examples of Big Data tools in HEP.

But before this..

Young collaborators from the Baltics

Vidmantas Zemleris <vidmantas.zemleris@gmail.com>

- work on DAS keyword search interface (work towards his master degree at EPFL)

Kipras Kančys <kipras.kan@gmail.com>

- work on dataset-popularity and SparkML interface for DCAFPilot project (during his Spring semester)

Aurimas Repečka <aurimas.repecka@mif.stud.vu.lt>

- work on PhEDEx replica monitoring tools (CERN summer student)

Zygimantas Matonis <mr.matonis@gmail.com>

- work on PhEDEx replica monitoring predictions (CERN summer student)

Adelina Varatinskaite <adelina.varatinskaite@mif.stud.vu.lt>

- work in DMWM on python2 to python3 migration (CERN summer student)

Excellent feedback by their CMS supervisors: *solid knowledge, hard workers, ability to quickly pick up new tools and get familiar with the sw environment, etc.*

CMS is looking forward to have more collaborators like you on board!

Skills set

Apart from the ⚐ *Tags (right-hand side of all projects slides in this talk)*, for guidance to both students and professors, the basic required skill set roughly covers:

- familiarity with Linux/UNIX tools
- Linux OS
- python
- Git, Javascript, web-development
- data-structures/algorithms
- Machine Learning
- BigData tools is a plus (e.g. Hadoop, HDFS, Spark)
- database(s) is a plus (ORACLE, MySQL, MongoDB)

References

[1] https://github.com/dmwm/DMWMAnalytics/tree/master/Popularity/DCAFPilot

[2] V. Kuznetsov, D. Bonacorsi et al, "*Predicting dataset popularity for the CMS experiment*", https://arxiv.org/abs/1602.07226v1 [Feb 2016]

[3] https://github.com/vkuznet/FTS-and-ML

[4] http://cern.ch/go/bf8Z, "***IT Monitoring Service***" [Nov 2016]

[5] http://cern.ch/go/6qqF, "***Usage of Big Data tools for HEP***" [Nov 2016]

[6] https://github.com/dmwm/PhedexReplicaMonitoring

[7] https://github.com/dmwm/WMArchive,

https://github.com/dmwm/WMArchive/wiki/WMArchive-architecture

[8] https://github.com/vkuznet/WMArchiveAggregation

[9] http://cern.ch/go/7L8S, "**Web concurrency R&D**" [Nov 2016],

http://cern.ch/go/N6wx,"**CMSWeb scalability issues**" [Nov 2016]

[10] https://github.com/vkuznet/DLEventClassification

Back-up

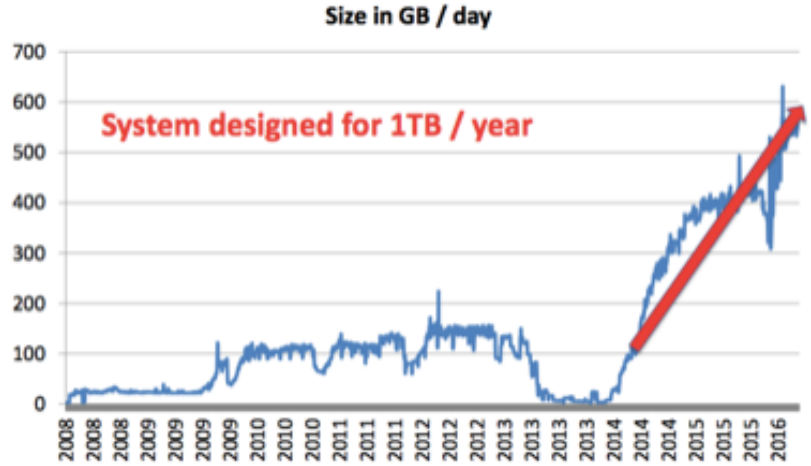
Example: **CERN Accelerator Logging Service**

CALS is used to store and retrieve billions of data acquisitions per day, from across the complete CERN accelerator complex, related sub-systems, and experiments

- launched in 2001, estimated to log 1 TB/year during LHC operation (>25 years)
- e.g. data logging proved useful to tune the SPS-to-LHC beam extraction process. Data logging capabilities started to be exploited for more systems
- *Today*: persistency layer has **~2M variables**, **1.6E12 data-points/year**, CALS records **6M extraction requests per day**, its storage needs easily exceed **0.5 TB/day**, LHC log data archived forever

Critical service for running the LHC (and not only)

- Constant performance improvements
- Decision support system for mgmt



[credits: D. Bonacorsi, presentation on "Big Data in Big Science at CERN" to Accenture [Sep 2016]

CALS (continued)

CALS successful for many years. As the accelerators have evolved and matured - logging and analysis needs to evolve too.

- Provide **analytics** functionalities with right tools for the job (**Big Data toolset**). Increase **bandwidth** & **processing power**
- Oracle set-up stretched to its very limit. Very extensive expertise required currently. Allow for **better API integration** with outside community (**Python**)

Work in progress on PoC for CALS 2.0 with Big Data toolset

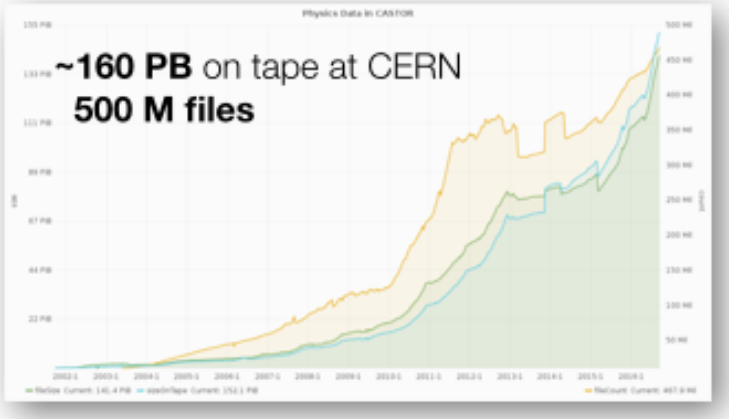
Currently exploring:

- **Hadoop** ecosystem at large
- Data formats: data comes as **Avro**, compactors convert to **Parquet**
- Data ingestion: serialize and send asynchronously to Apache **Kafka**, separate streams to (LinkedIn) **Gobblin** as ETL framework
- Data extraction: (Cloudera) **Impala** as query engine and **Spark** as distributed data processing framework

VALUE of Big Data toolset: allow a critical system to scale up, enabling exploration of bigger data sets at same (or improved) performances

[credits: D. Bonacorsi, presentation on "Big Data in Big Science at CERN" to Accenture [Sep 2016]

Example: **Tape Storage event logging**



CASTOR is used at CERN as a Mass Storage System (MSS)

Metadata are cumulative upon writing/reading data on tapes

- on which tape is my file stored? Is there also a copy on disk? List all files in a given tape. Was the tape repacked? Which read/write errors we experienced on a tape since ever?
- cluster with ~10 servers, ~100 GB of logs per day archived

Flexible monitoring for tape operations at CERN

- data fetched from **HBase** in fractions of sec

VALUE of Big Data toolset: improve CERN tape operations and reduce debugging time through reliable and performant access to metadata on tape usage

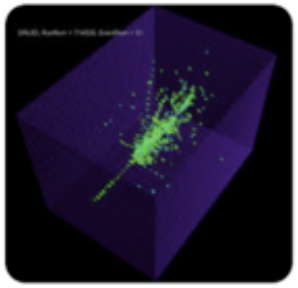
[credits: D. Bonacorsi, presentation on "Big Data in Big Science at CERN" to Accenture [Sep 2016]

Example(s): **Data reconstruction algorithms**

New algorithms don't need to be "better", they need to be **FASTER**

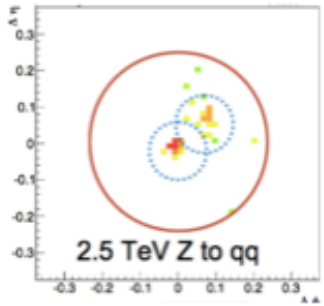
Example 1: calorimeter pattern recognition (e.g. CMS)

- particles emerging from collisions are brought to a stop in calorimeters
- showers of fragments (particles) wih depth, intensity, topology characteristic of each particle type and kinematics. Timing of energy deposition is also used
- accuracy is correlated with the granularity of the detector, and conventional algorithm cannot cope with the increase granularity of next generation ones



Example 2: boosted objects imaging

- decays of high momentum particles are boosted along their initial direction
- existing techniques to disambiguate fall short on dense 'jet' cases with many overlapping particles
- identification at the level of event filtering at high rate is impossible due to algorithm computation



Particle identification with **deep learning pattern recognition** is showing promising results

Great support by **CERN openlab** in matching with industry partners

VALUE of deep learning: in synergy with industry, computation acceleration and access to deep training facility can allow to attack these challenges

[credits: D. Bonacorsi, presentation on "Big Data in Big Science at CERN" to Accenture [Sep 2016]