

Data Analytics and “Big Data” Platforms at CERN

Use Cases, Platforms, Services

July 3rd, 2018

Luca Canali – Analytics and Streaming Service

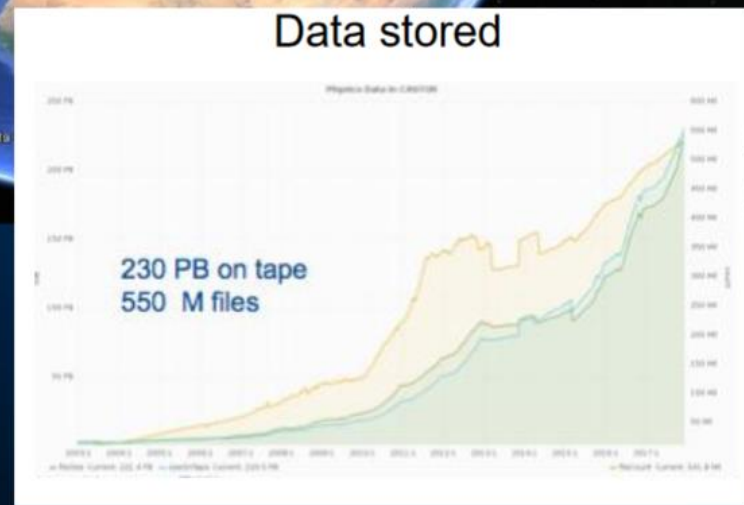
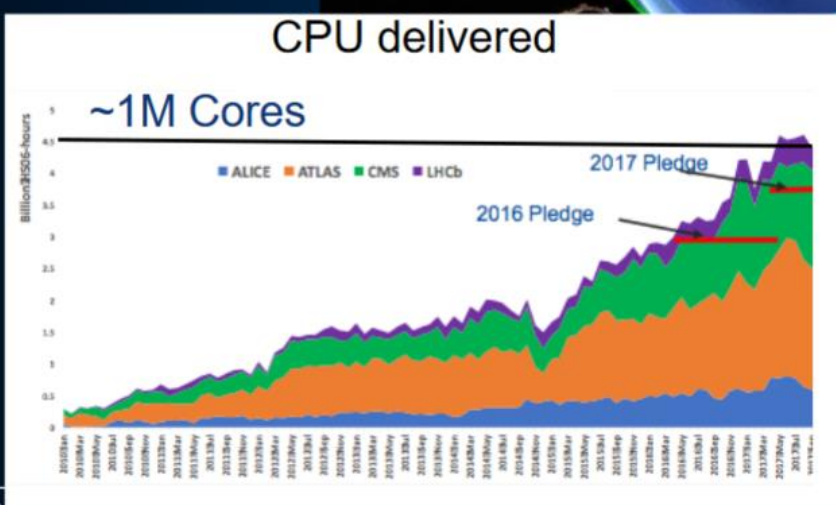
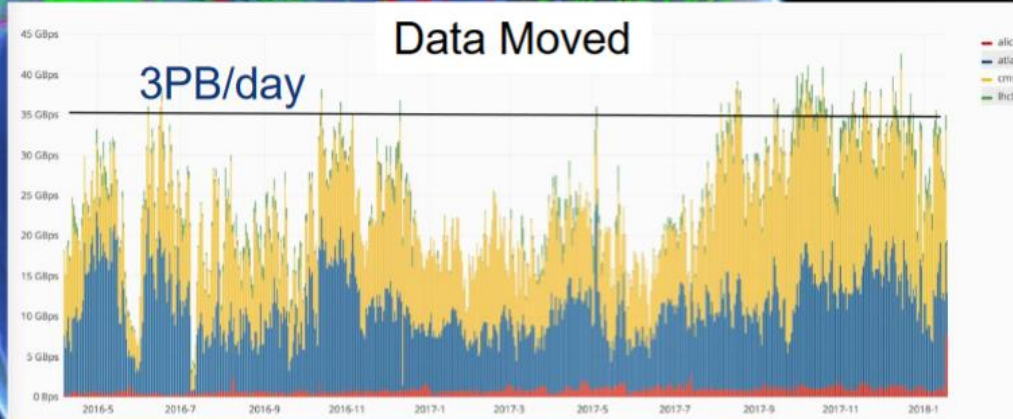
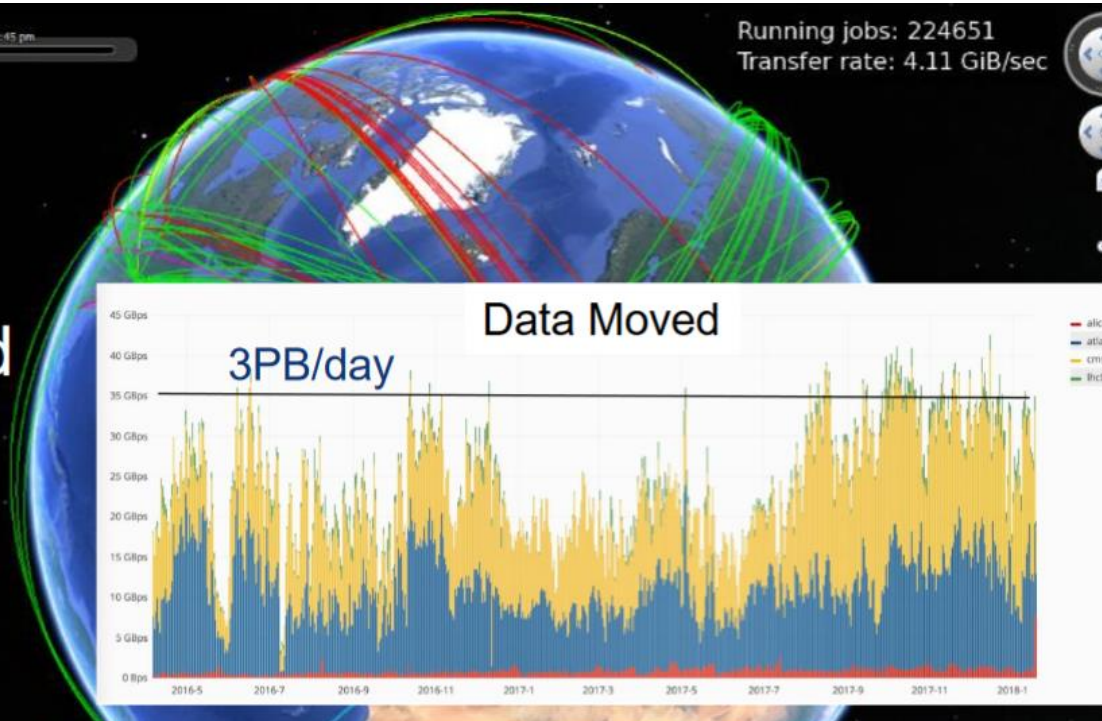
Data at Scale @CERN

- Physics data – Today we use WLCG to handle it
 - Optimised for physics analysis and concurrent access
 - ROOT framework - custom software and data format
 - Early stage experimental work ongoing to use Spark for physics analysis
- Infrastructure data – Industry and open source “big data” tools are widely used
 - Accelerators and detector controllers
 - Experiments Data catalogues (collisions, files etc.)
 - Monitoring of the WLCG and CERN data centres
 - Systems logs



LHC Data

- Worldwide distribution and processing of LHC data

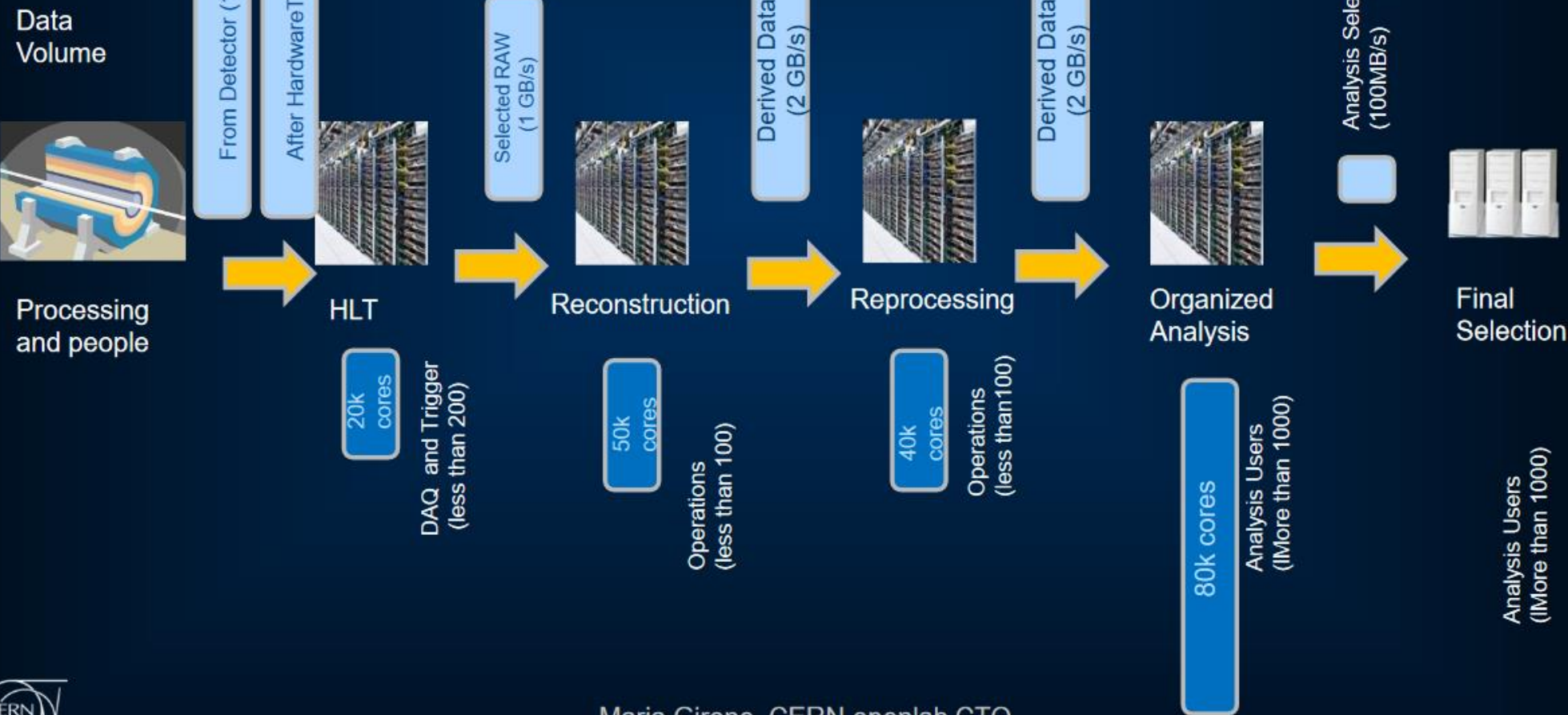


LHC data processing teams have built custom solutions able to operate at very large scale



The process to transform raw data into useful physics datasets

- This is a complicated series of steps at the LHC



Complex pipelines for data analysis at LHC

Build on many years of experience and world-wide collaborations

Hadoop, Spark and Kafka Services at CERN

- Setup and run the infrastructure for scale-out analytics solutions
- Today primarily for the components from Apache Hadoop framework and Big Data Ecosystem
- Support user community
 - Provide consultancy
 - Ensure knowledge sharing
 - Train on the technologies
 - Build the community



“Big Data” on Hadoop Clusters at CERN

- Several orders of magnitude below LHC data processing systems
- 3 current production Hadoop clusters
 - + environments for NXCALS DEV and HadoopQA
 - Just commissioned a new system for **BE NXCALS** (accelerator logging) platform
- Numbers relate to the size of the infrastructure (updated Q2 2018):
 - 14 PB Storage, 110 nodes, 3100 logical cores, 20 TB memory

Hadoop and Spark Clusters

- Software – mixture of CERN Apache Hadoop and Cloudera Distribution for Hadoop

| Cluster Name | Configuration | Software Version |
|-----------------------------|--|------------------|
| Accelerator logging, NXCALS | 20 nodes (Cores 480, Mem - 8 TB, Storage – 5 PB, 96GB in SSD) | hadoop_cern |
| General Purpose | 48 nodes (Cores – 892, Mem – 7.5TB, Storage – 6 PB) | cdh |
| Development cluster | 14 nodes (Cores – 196, Mem – 768GB, Storage – 2.15 PB) | cdh |
| ATLAS Event Index | 18 nodes (Cores – 288, Mem – 912GB, Storage – 1.29 PB) | cdh |
| QA cluster | 10 nodes | hadoop_cern |

Analytics Pipelines – Use Cases

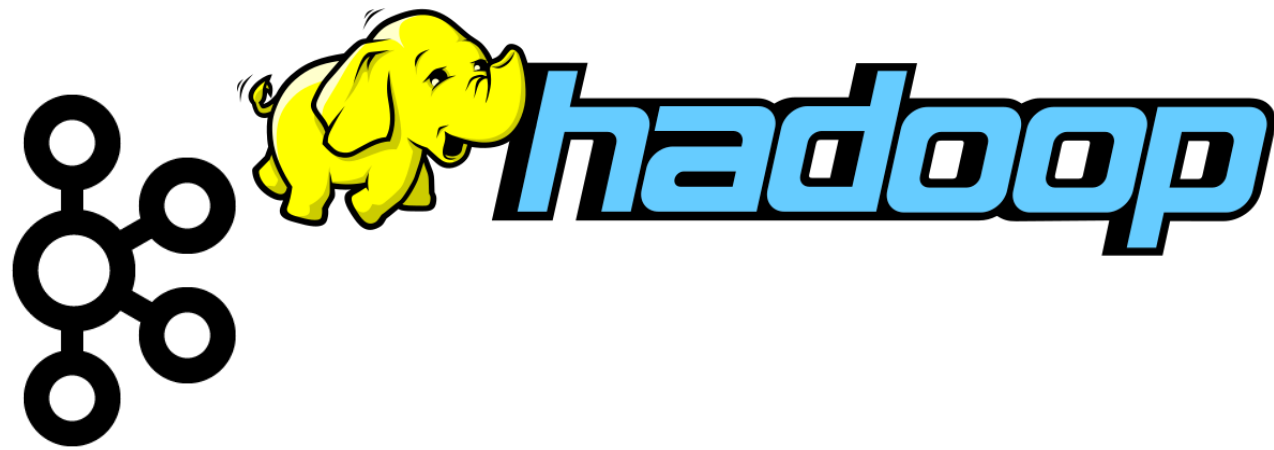
- Many use cases at CERN for analytics
 - Data analysis, dashboards, plots, joining and aggregating multiple data, libraries for specialized processing, machine learning, ...
- Communities
 - **Physics:**
 - Analysis on computing metadata (e.g. studies of popularity, grid jobs, etc) (CMS, ATLAS)
 - Development of new ways to process **ROOT** data, e.g.: data reduction and analysis with Spark-ROOT by CMS Bigdata project, also TOTEM working on this
 - **IT:**
 - Analytics on IT monitoring data
 - Computer security
 - **BE:**
 - NX CALS – next generation accelerator logging platform
 - BE controls data and analytics
 - More:
 - Many tools provided in our platforms are popular and readily available, likely to attract **new** projects, notably the analytics platform with hosted notebooks **SWAN_Spark**
 - E.g. Starting investigations on data pipelines for IoT (Internet of Things)

“Big Data”: Not Only Analytics

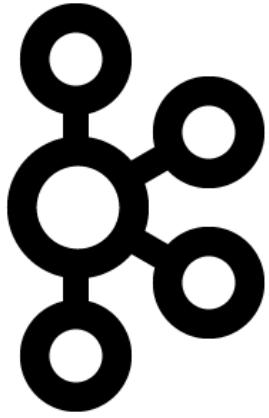
- Data **analytics** is a key use case for the platforms
- Scalable workloads and parallel **computing**
 - Example work on data reduction (CMS Big Data project) and parallel processing of ROOT data (EP-SFT)
- **Database**-type workload also important
 - Use Big Data tools instead of RDBMS
 - Examples: NXCALS, ATLAS EventIndex, explorations on WINCC/PVSS next generation
- Data pipelines and **streaming**
 - See example of monitoring and Computer security (Kafka development with help of CM)
 - Also current investigations on IoT (project with CS)

Highlights of “Big Data” Components

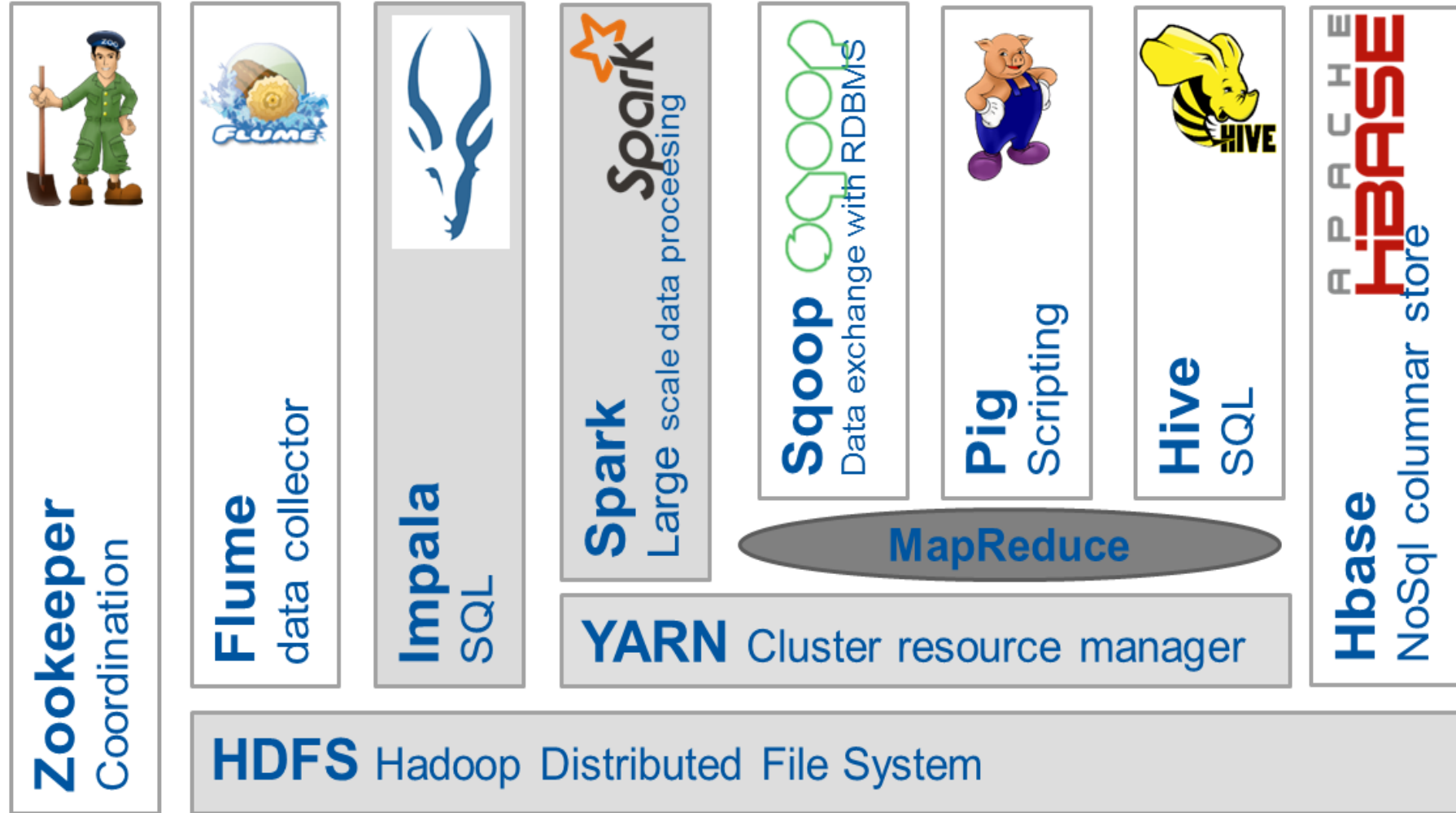
- Apache Hadoop clusters with YARN and HDFS
 - Also HBase, Impala, Hive,...
- Apache Spark for analytics
 - Apache Kafka for streaming
- Data: Parquet, JSON, ROOT
- UI: Notebooks/ SWAN



Overview of available components in 2018



Kafka:
streaming
and ingestion

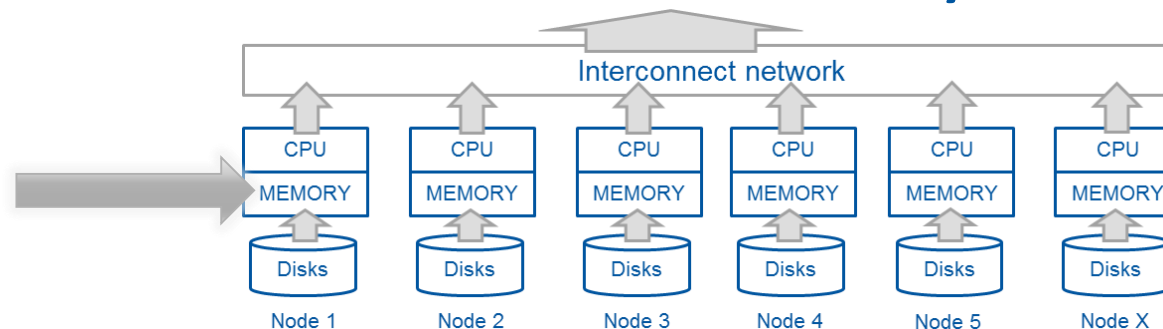


Hadoop and Spark for big data analytics



- Distributed systems for data processing
 - Both imperative and declarative programming interfaces
 - Can operate at scale by design (shared nothing)
 - Typically on clusters of commodity-type servers
 - Many solutions target data analytics and data warehousing
 - Can do much more: stream processing, machine learning
- Already well established in the industry and open source

Scale-out data processing



Hadoop and Spark production deployment

- Software distribution
 - Cloudera (since 2013)
 - Vanilla Apache (since 2017)



- Rolling change deployment
 - no service downtime
 - transparent in most of the cases



- Installation and configuration
 - CentOS 7.4
 - custom Puppet module



- Host monitoring and alerting
 - via CERN IT Monitoring infrastructure



- Security
 - authentication Kerberos
 - fine-grained authorization integrated with e-groups



- Service level monitoring
 - metrics integrated with: Elastic + Grafana
 - custom scripts for availability and alerting



- High availability
 - automatic master failover for HDFS, YARN and HBASE

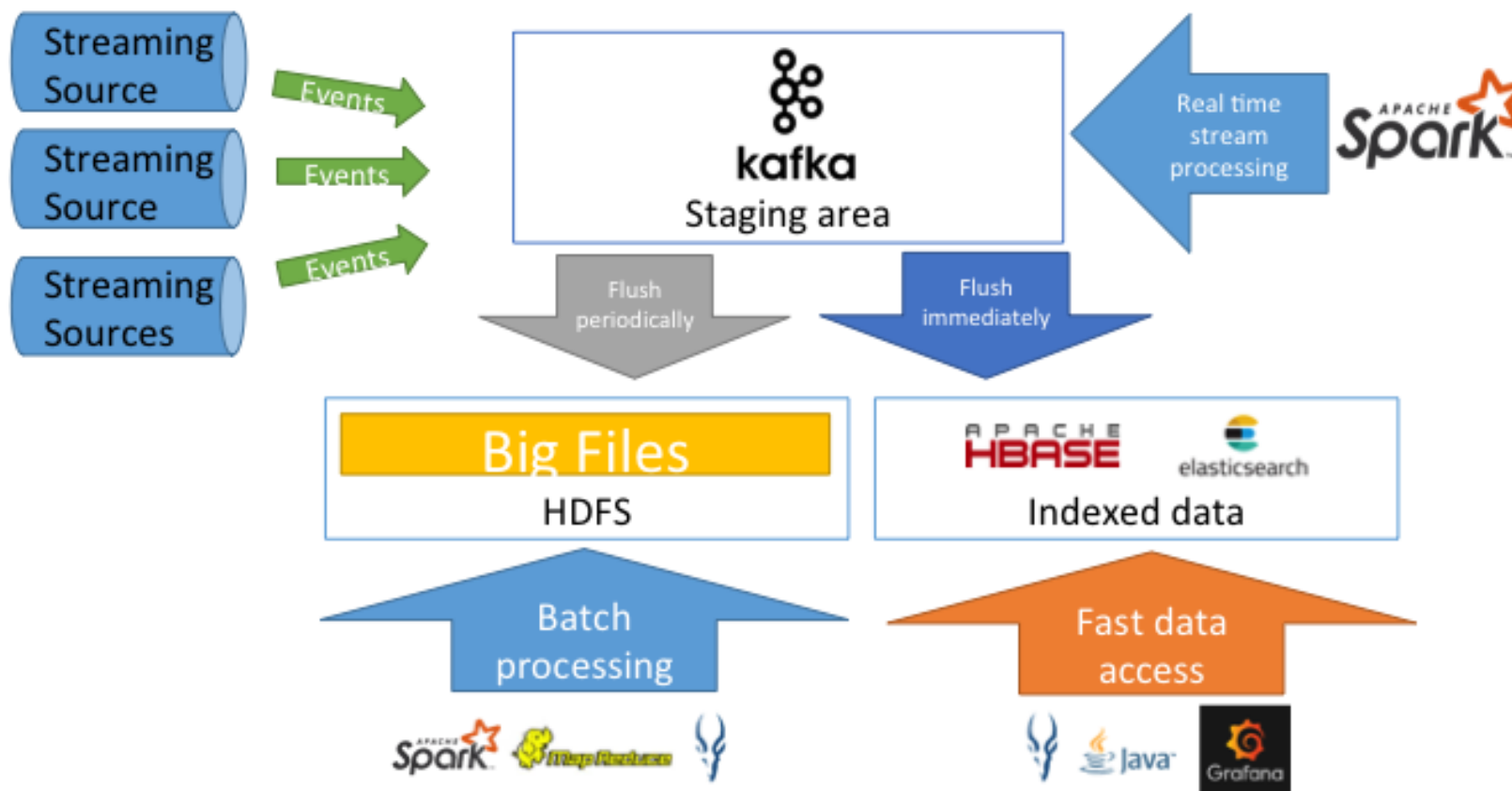


- HDFS backups
 - Daily incremental snapshots
 - Sent to tapes (CASTOR)



Apache Kafka - data streaming at scale

- Apache Kafka streaming platform is a standard component for modern scalable architectures
- Started providing Kafka as a pilot service in 2017
- Current development: Kafka for IoT data ingestion platform



SWAN – Jupyter Notebooks On Demand



- SWAN - Service for Web based Analysis
 - Developed at CERN, provides Jupyter notebooks on demand with relevant CERN integration for data and compute
- An interactive platform that combines code, equations, text and visualizations
 - Ideal for exploration, reproducibility, collaboration
- Fully integrated with Spark and Hadoop clusters at CERN
 - Python on Spark (PySpark) at scale
 - Modern, powerful and scalable platform for data analysis
 - Web-based: no need to install any software



Do the heavylifting in spark and collect aggregated view to panda DF

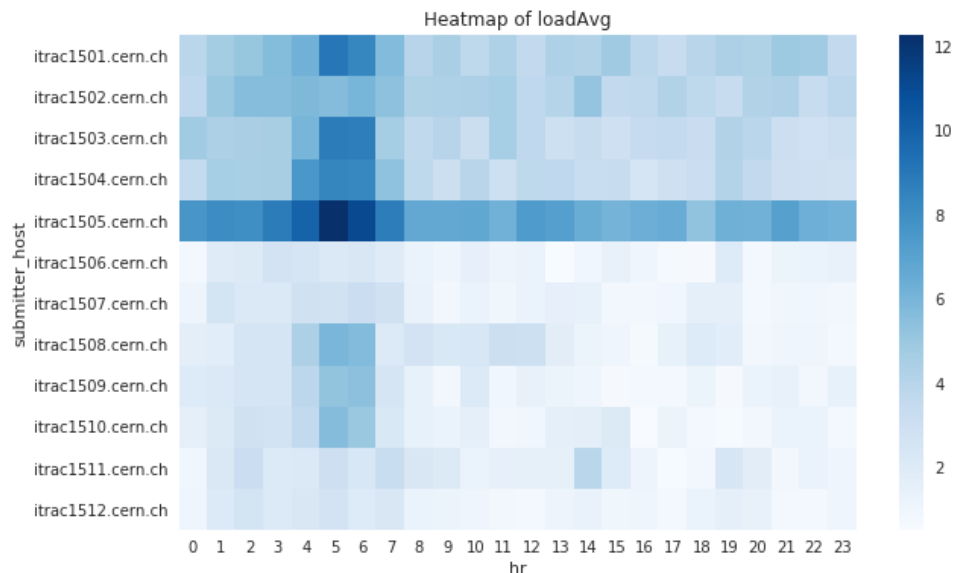
```
In [11]: df_loadAvg_pandas = spark.sql("SELECT submitter_host, \
    avg(body.LoadAvg) as avg, \
    hour(from_unixtime(timestamp / 1000, 'yyyy-MM-dd HH:mm:ss')) as hr \
    FROM loadAvg \
    WHERE submitter_hostgroup = 'hadoop/itdb/datanode' \
    AND dayofmonth(from_unixtime(timestamp / 1000, 'yyyy-MM-dd HH:mm:ss')) = 15 \
    GROUP BY hour(from_unixtime(timestamp / 1000, 'yyyy-MM-dd HH:mm:ss'), submitter_host")\
    .toPandas()
```

| Job ID | Job Name | Status | Stages | Tasks | Submission Time | Duration |
|--------|----------|-----------|--------|-----------|-----------------|----------|
| 3 | toPandas | COMPLETED | 2/2 | 388 / 388 | 4 minutes ago | 36s |

Visualize with seaborn

```
In [19]: # heatmap of service availability
plt.figure(figsize=(10, 6))
ax = sns.heatmap(df_loadAvg_pandas.pivot(index='submitter_host', columns='hr', values='avg'), cmap="Blues")
ax.set_title("Heatmap of loadAvg")
```

Out[19]: Text(0.5,1,u'Heatmap of loadAvg')



Text

Code

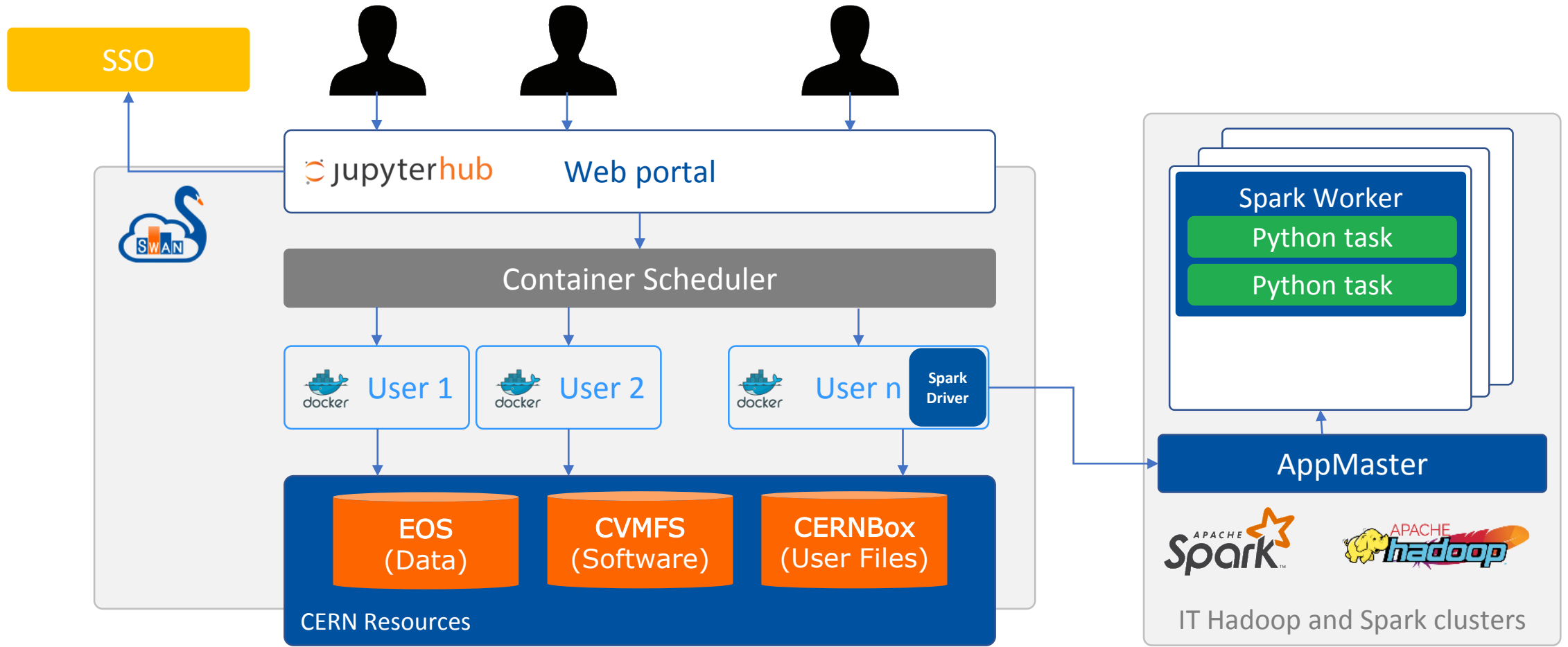
Monitoring

Visualizations

All the required tools, software and data available in the single window!



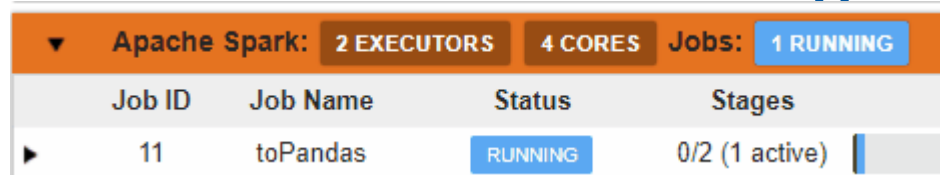
SWAN – Architecture



Integration of Spark with Jupyter notebooks

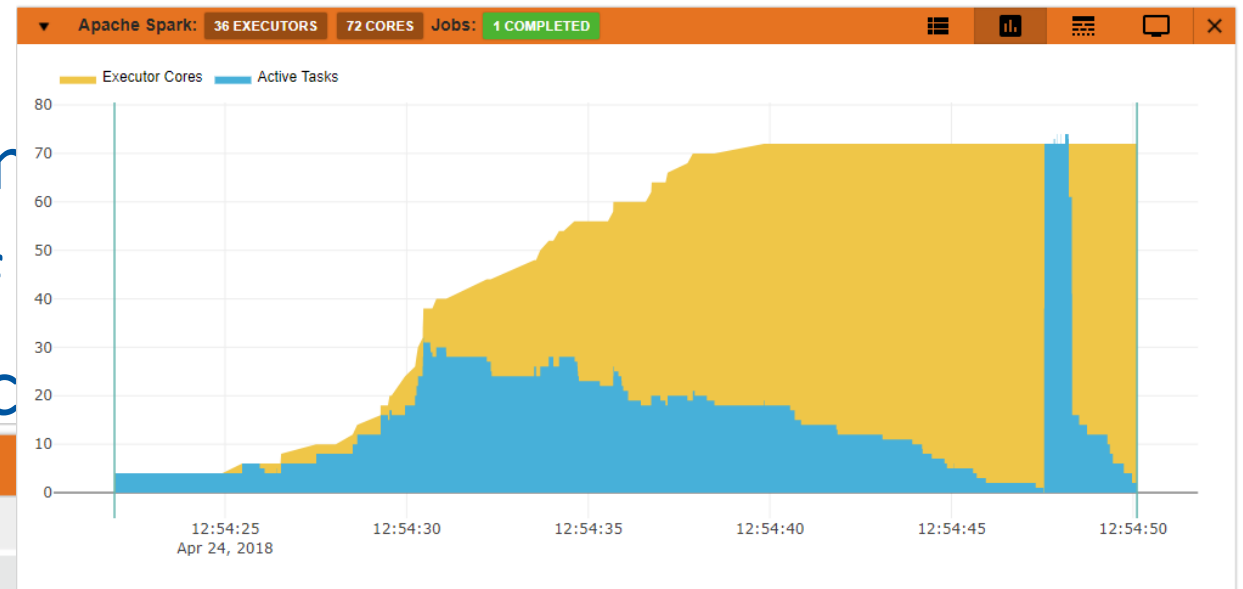
- SparkConnector – python module handling Spark Application configuration complexity

- Spark Monitor – jupyter r
- For live monitoring of notebook including ac



Apache Spark: 2 EXECUTORS 4 CORES Jobs: 1 RUNNING

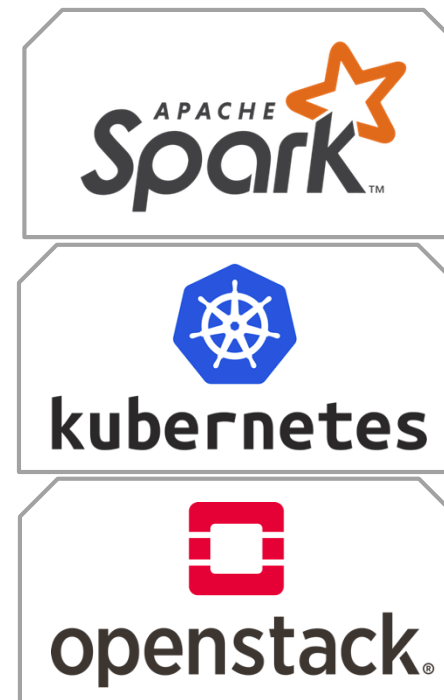
| Job ID | Job Name | Status | Stages |
|--------|----------|---------|----------------|
| 11 | toPandas | RUNNING | 0/2 (1 active) |



- Authentication and Encrypting – All the actors in the spark application are authenticated using shared secret

Spark as a service on private cloud

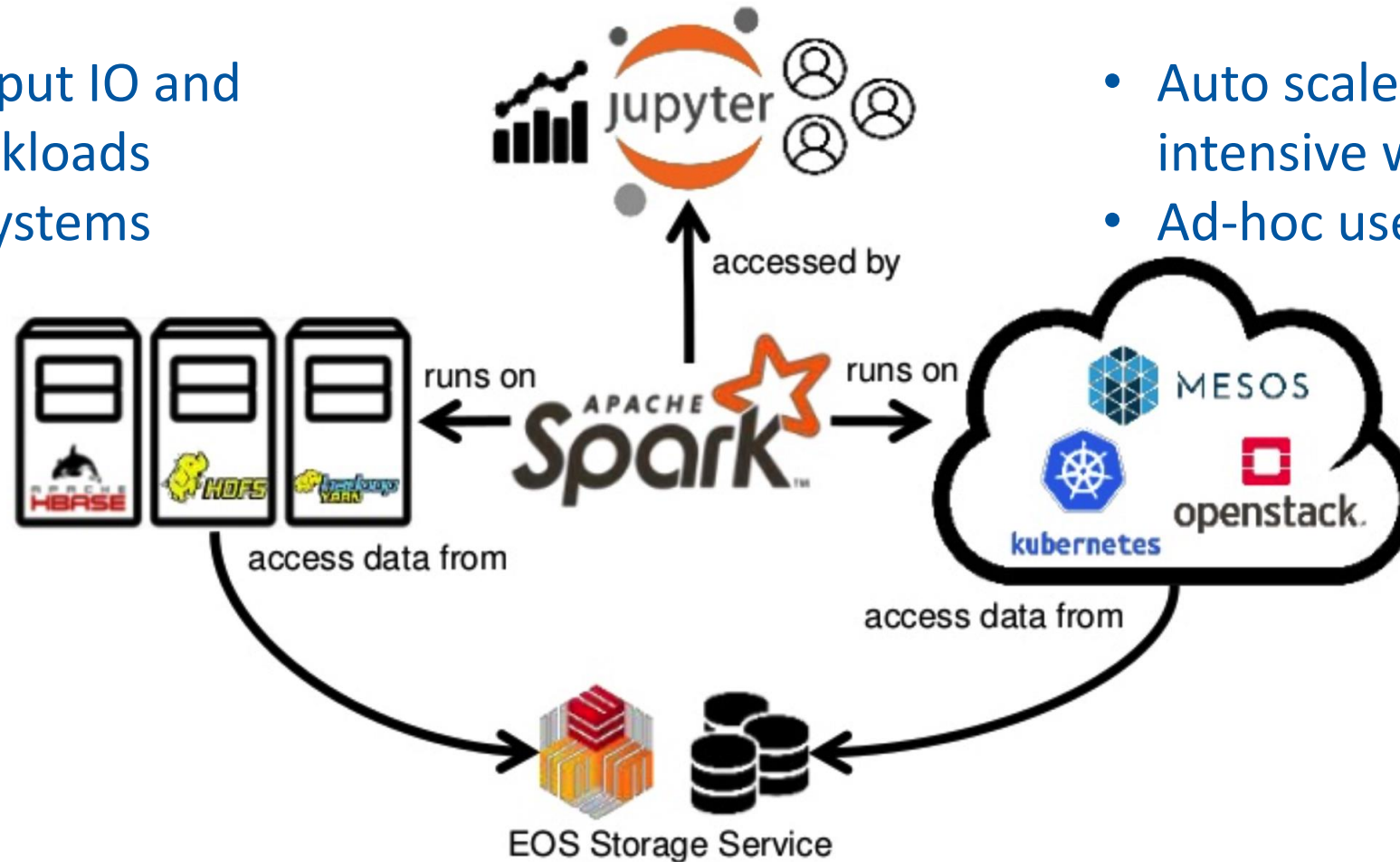
- Ongoing work to provide Spark on Kubernetes clusters on CERN private cloud
- Promising solution to adapt Spark to cloud native model
 - Processing of data that resides in external storages (non-HDFS)
 - Auto scale compute resources depending on needs to lower costs
- Spark clusters on containers
 - Kubernetes over Openstack
 - Leveraging the Kubernetes support in Spark 2.3
- Initial Use cases
 - Ad-hoc users with high demand computing resource demanding workloads
 - Streaming workloads (e.g. accessing Apache Kafka)



Analytics platform outlook

- High throughput IO and compute workloads
- Established systems

- Auto scale for compute intensive workloads
- Ad-hoc users



Engineering Efforts to Enable Effective ML

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

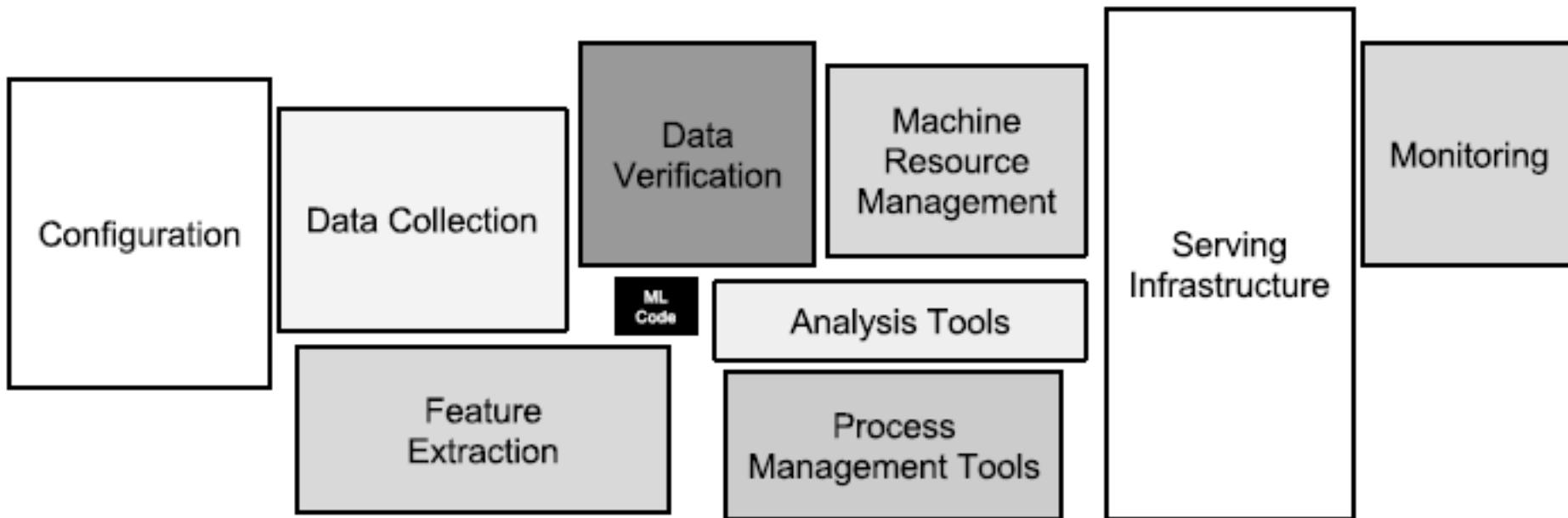
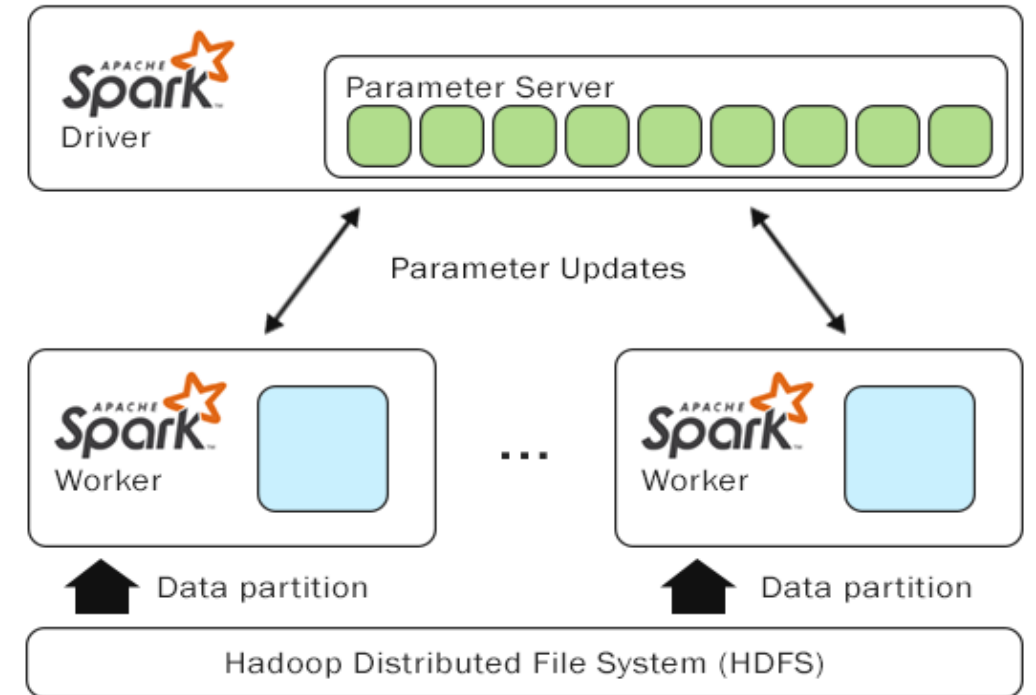


Figure 1: Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle. The required surrounding infrastructure is vast and complex.

Machine Learning with Spark

- Spark has tools for **machine learning at scale**
 - Spark library MLlib
- Distributed deep learning
 - Working on use cases with CMS and ATLAS
 - We have developed an integration of Keras with Spark
- Tests and future investigations:
 - Frameworks and tools for distributed deep learning with Spark available on open source:
 - TensorFlow, BigDL, TensorFlowonSpark, DL4j, ..
 - Also of interest HW solutions: for example FPGAs, GPUs etc

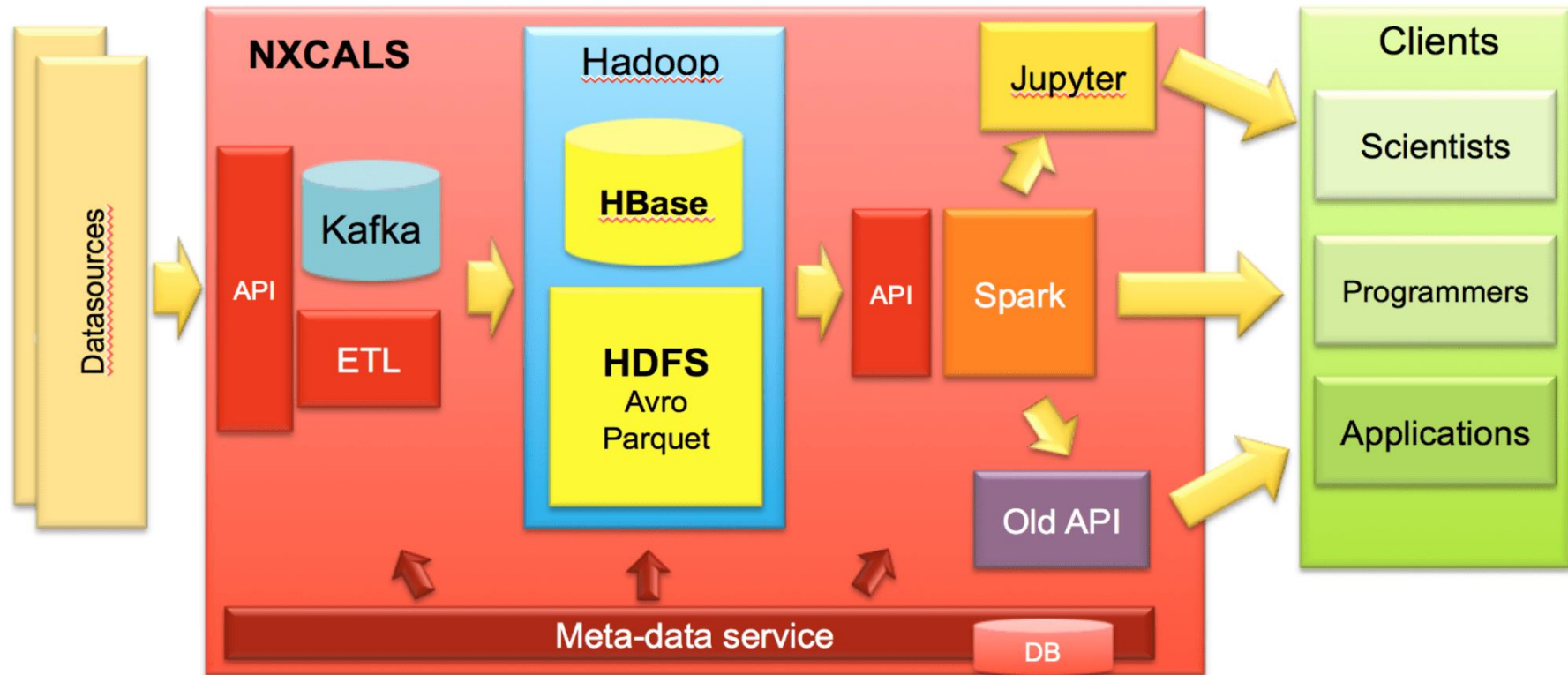


<https://github.com/cerndb/dist-keras>

Selected “Big Data” Projects at CERN

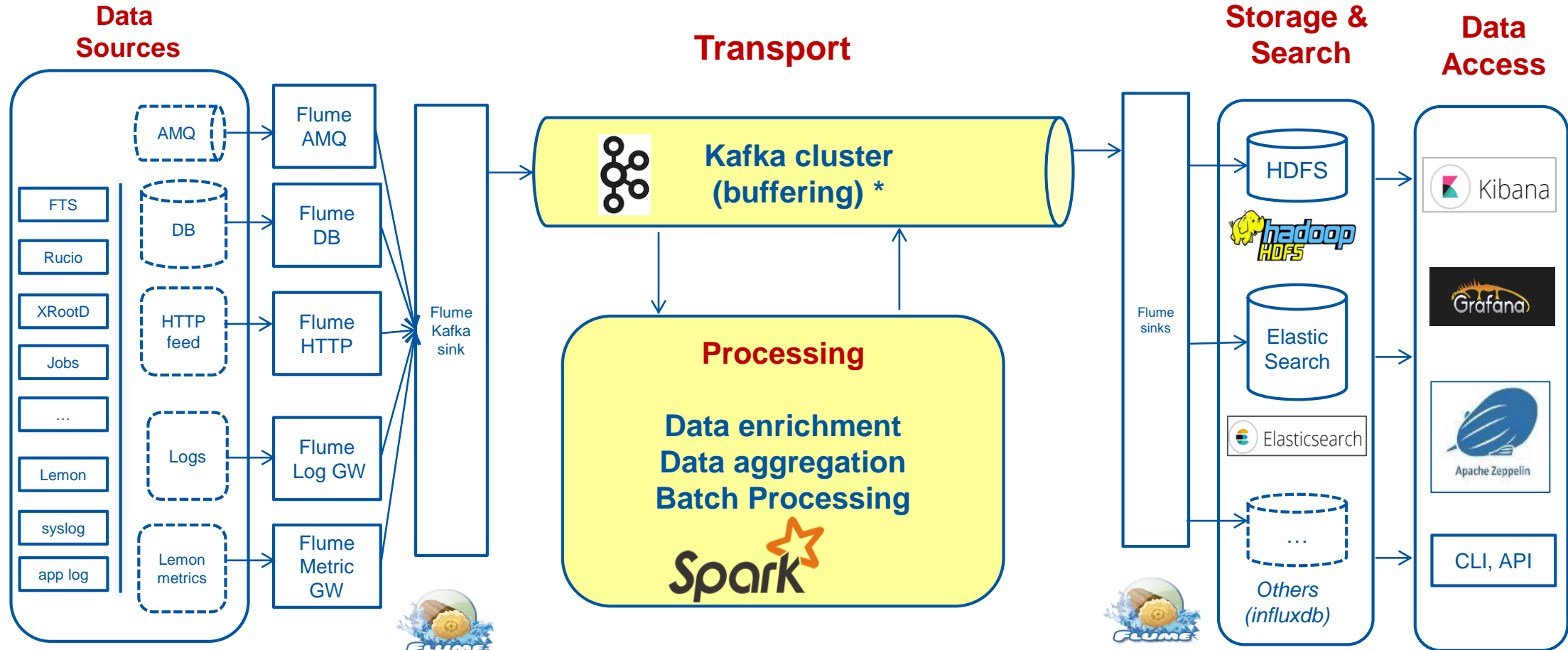
Next Gen. CERN Accelerator Logging

- A control system with: Streaming, Online System, API for Data Extraction
- Critical system for running **LHC - 700 TB today, growing 200 TB/year**



New CERN IT Monitoring infrastructure

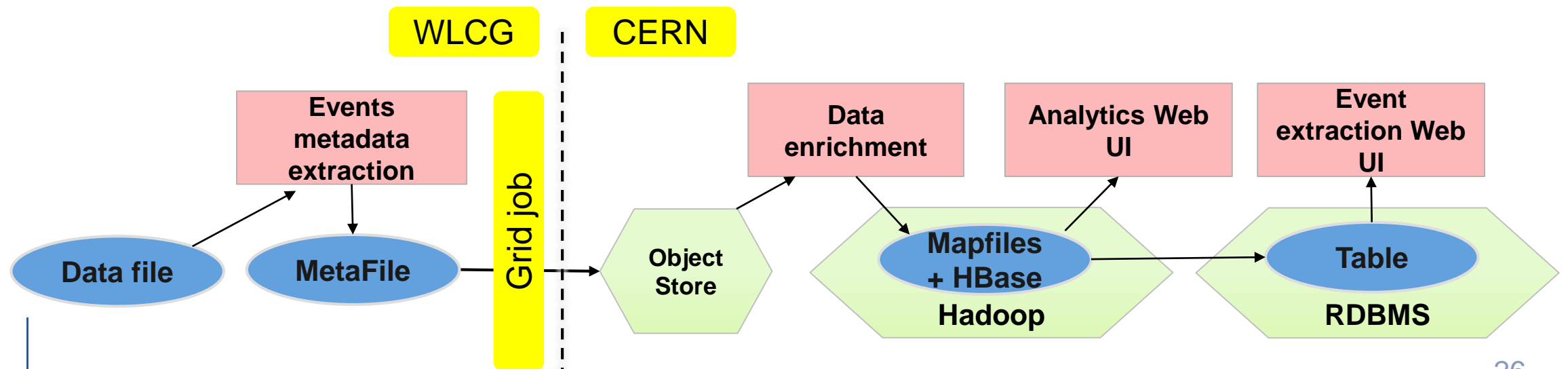
Critical for CC operations and **WLCG**



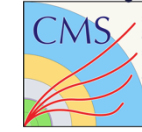
- Data now 200 GB/day, 200M events/day
- At scale **500 GB/day**
- Proved to be effective in several occasions

The ATLAS EventIndex

- Catalogue of all collisions in the ATLAS detector
 - Over 120 billion of records, 150TB of data
 - Current ingestion rates 5kHz, 60TB/year

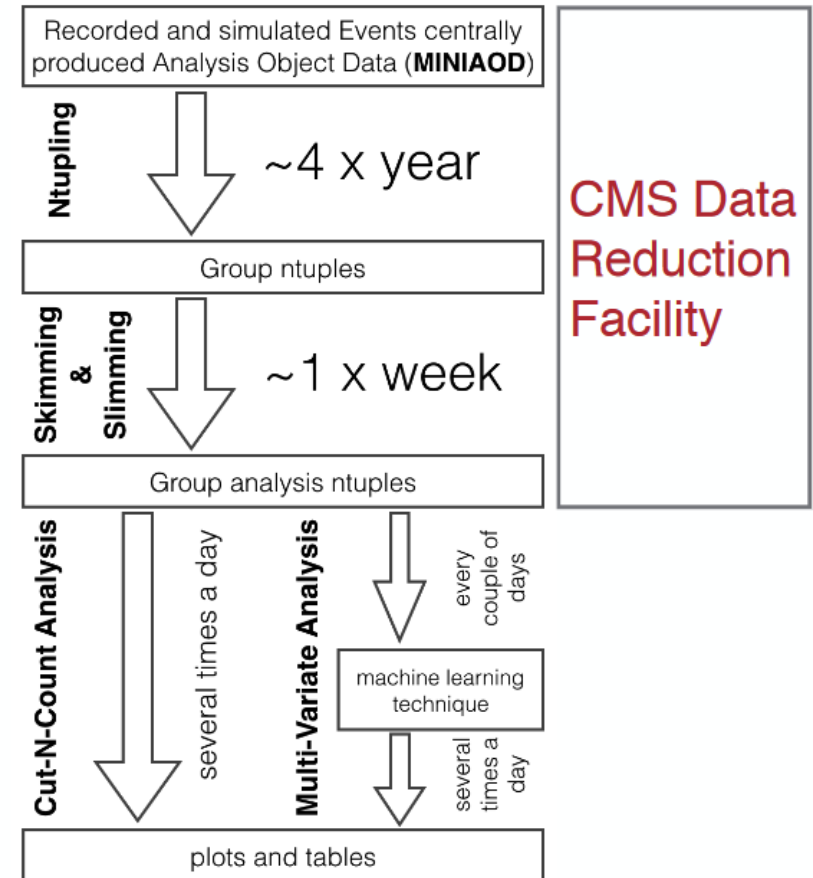


CMS Data Reduction Facility



- CMS Bigdata project, CERN openlab, Intel:
 - Reduce time to physics for PB-sized datasets
 - Exploring a possible new way to do HEP analysis
 - Improve computing resource utilization
 - Enable physicists to use tools and methods from “Big Data” and open source communities
- CMS Data Reduction Facility:
 - Goal: produce reduced data n-tuples for analysis in a more agile way than current methods
 - Currently testing: scale up with larger data sets, first prototype of reducing 1TB dataset is completed

Experimental and Ongoing R&D



R&D: Data Analysis with PySpark for HEP

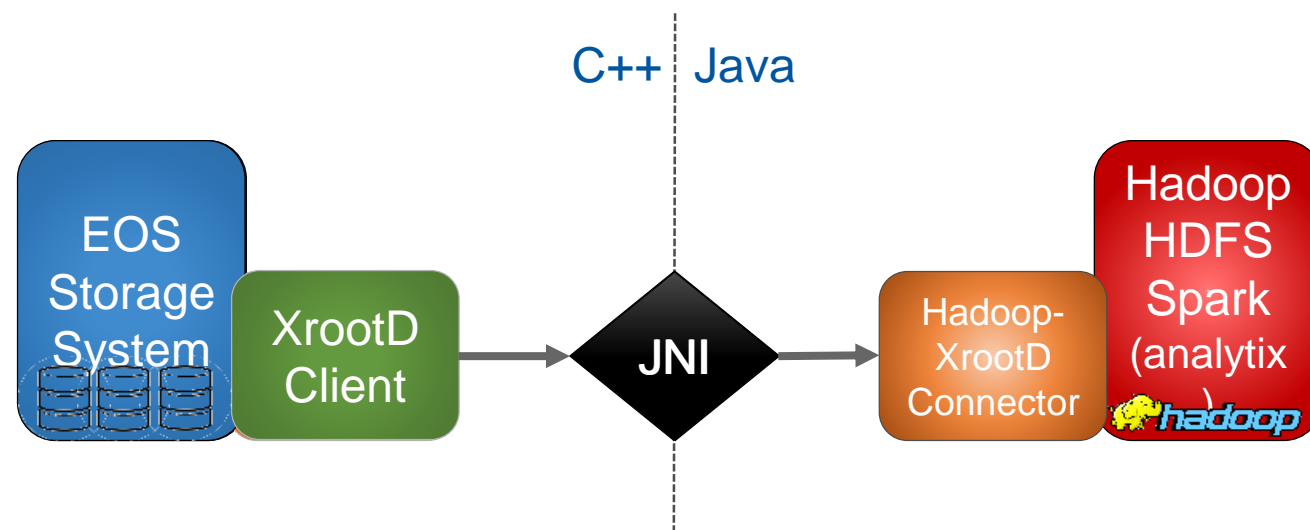


- Data access – **XRootD**
- Reading root files – **spark-root**
- User interface - **SWAN**



XRootD connector for Hadoop and Spark

- A library that binds Hadoop-based file system API with XRootD native client
 - Developed by CERN IT
- Allows most of components from Hadoop stack (**Spark**, MapReduce, Hive etc) to read from/write to EOS and CASTOR **directly**
 - Works with Grid certificates and Kerberos for authentication
 - Used for: HDFS backups, performing analytics on data stored on EOS (CERNBox)



Data Ingestion: spark-root

0.1.16 available on Maven Central!

- spark-root - ROOT I/O for JVM
- Extends Apache Spark's Data Source API
- Maps each ROOT TTree to Dataset[Row]
- Parallelization = # ROOT files.

Credits: V. Khristenko, J. Pivarski, diana-hep and CMS Big Data project



Scala

```
// inject the Dataset[Row]
import org.dianahep.sparkroot.experimental._
val df = spark.read.option("tree", <treeName>).root("<path/to/file>")

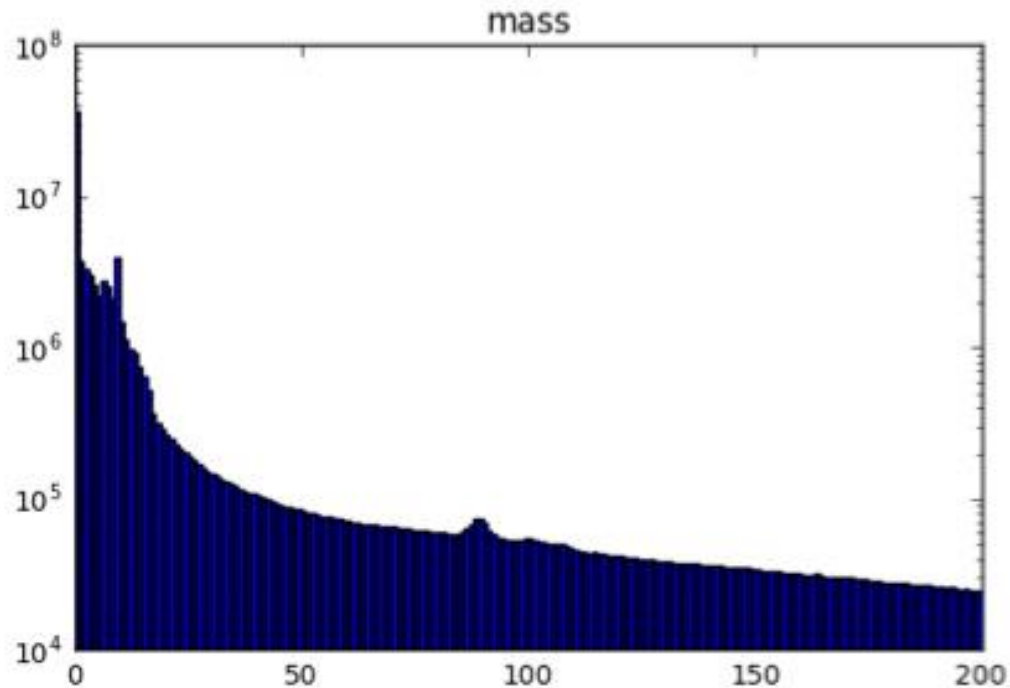
// pretty print of the schema
df.printSchema

|-- Particle: array (nullable = true)
|   |-- element: struct (containsNull = true)
|   |   |-- fUniqueID: integer (nullable = true)
|   |   |-- fBits: integer (nullable = true)
|   |   |-- PID: integer (nullable = true)
|   |   |-- Status: integer (nullable = true)
|   |   |-- IsPU: integer (nullable = true)
|   |   |-- M1: integer (nullable = true)
|   |   |-- M2: integer (nullable = true)
|   |   |-- D1: integer (nullable = true)
|   |   |-- D2: integer (nullable = true)
|   |   |-- Charge: integer (nullable = true)
|   |   |-- Mass: float (nullable = true)
|   |   |-- E: float (nullable = true)
```

Data Processing: CMS Open Data Example

Let's calculate the invariant mass of a di-muon system?!

- Transform a collection of muons to an invariant mass for each Row (Event).
- Aggregate (histogram) over the entire dataset.



```
# read in the data
df = sqlContext.read\
    .format("org.dianahep.sparkroot.experimental")\
    .load("hdfs:/path/to/files/*.root")

# count the number of rows:
df.count()

# select only muons
muons =
df.select("patMuons_slimmedMuons__RECO_.patMuons_slimmedMuons__RECO_obj.m_state").toDF("muons")

# map each event to an invariant mass
inv_masses = muons.rdd.map(toInvMass)

# Use histogrammar to perform aggregations
empty = histogrammar.Bin(200, 0, 200, lambda row: row.mass)
h_inv_masses = inv_masses.aggregate(empty,
    histogrammar.increment,
    histogrammar.combine)
```

Summary

- Demand of “Big Data” platforms and tools is growing at CERN
 - Many successful projects in production
 - Critical NXCals project – for the smooth running of LHC is in early production and ramp up phase
 - Apache Spark is the main compute framework used by physicists and researchers
- Hadoop, Spark, Kafka services at CERN IT
 - Service is evolving: High availability, security, backups, external data sources, notebooks, cloud native workloads...
- Experience and Knowledge Transfer to Industry
 - Technologies evolve rapidly and knowledge sharing very important
 - We are happy to share our experience, tools, software and original work carried out at CERN

Future work

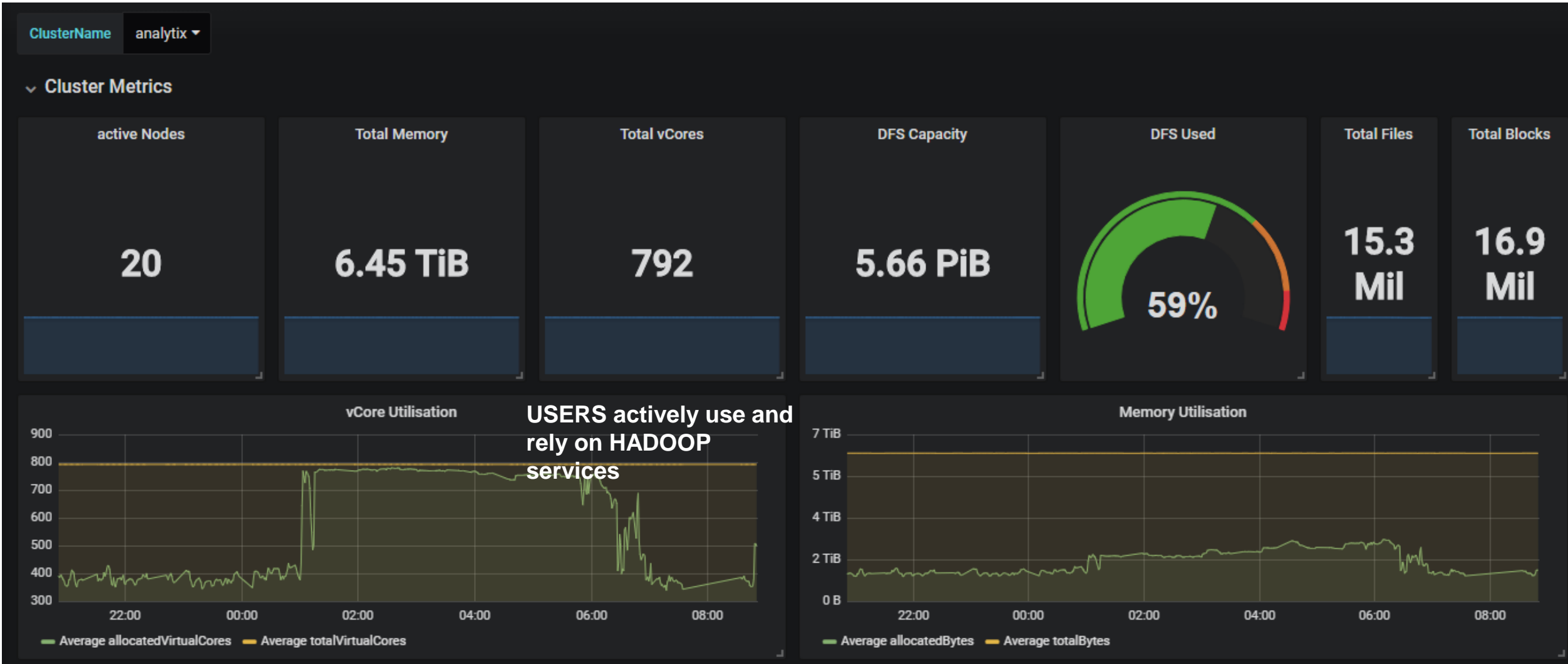
- Roll out of Spark on Kubernetes production service
- Roll out of Kafka service to full production state
- Continuing the work on optimizing physics analysis with Apache Spark
- Bringing additional functionality to SWAN (Analytics Platform)
 - Attaching GPU resources
 - Tighter integrations with other components of Hadoop stack
- Improvements to Hadoop service
 - Accounting of Resource usage, Streamlining resource requests, Monitoring and Alerting
- Further explore the big data technology landscape
 - Mainly Apache Kudu and Presto

Acknowledgements

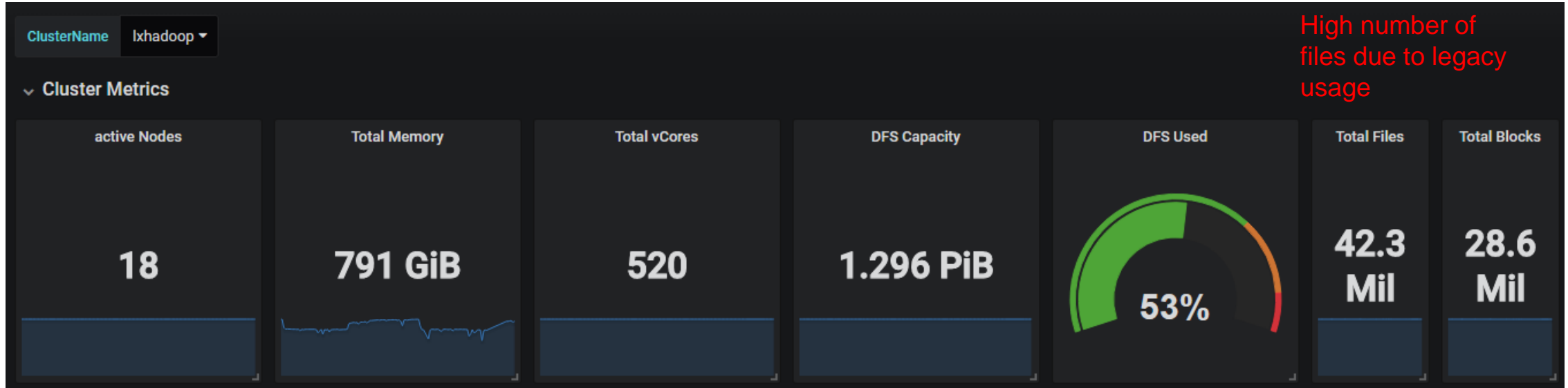
- CERN Colleagues in the Scalable Analytics Solutions section (IT-DB-SAS) at CERN
- CERN colleagues from EP-SFT and IT-ST on the joint development of SWAN
- Users of the analytics service, including IT Monitoring, IT Security, ATLAS DDM and EventIndex, CMSSpark project, Accelerator logging and controls
- CMS Big Data project, with Intel and CERN OpenLab

Backup Material

Hadoop and Spark Clusters Usage



Hadoop and Spark Clusters Usage



- Usage of Spark and Hadoop is growing at CERN
- Dealing with legacy issues
 - Correct usage of Hadoop stack
 - Teams working with snapshotted technologies from years ago (MapFiles?)
- Hadoop and Spark team adds real value in helping teams maximize their return on Hadoop and Spark investments

Moving to Apache Hadoop distribution (since 2017)

- Better control of the core software stack to adapt to CERN needs
 - Independent from a vendor/distributor
 - Enabling non-default features (compression algorithms, R for Spark)
 - Backporting critical and necessary patches
- We do rpm packaging for core components
 - HDFS and YARN, Spark, HBase
- Streamlined development
 - Available on Maven Central Repository



Challenges and Activities

- Platforms
 - Provide evolution for **HW** (Hadoop platform) and **SW** (distribute and update software and configuration)
- Service
 - Support **production** services (IT monitoring, Security, ATLAS EventIndex)
 - Build robust service for critical platform (NXCALS and more) using custom-integrated **open source** software solutions in constant **evolution**
 - Evolve service configuration and procedures to fulfil **users** needs
 - Further **grow value** for community and projects
 - SWAN and general efforts to build and grow a **data analysis platform**
- Knowledge, experience, community
 - Technology keeps evolving, need to learn and adapt quickly to **change**