

Hadoop, Spark and Kafka Service

Current usage of LCG Releases and CVMFS

Librarian Workshop, 30th May 2018

Prasanth Kothuri (IT-DB-SAS)

On behalf of Hadoop, Spark and Kafka Service

IT Hadoop, Spark and Kafka Service

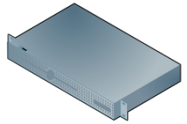
- Setup and run the infrastructure for scale-out analytics solutions
- Today primarily for the components from Apache Hadoop framework and Big Data Ecosystem
- Support user community
 - Provide consultancy
 - Ensure knowledge sharing
 - Train on the technologies
 - Build the community



Hadoop and Spark service in numbers



- ✧ 6 clusters
 - ✧ 4 production (bare-metal)
 - ✧ 2 QA clusters (VMs)



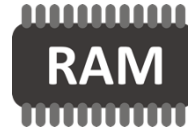
- ✧ 110+ physical servers



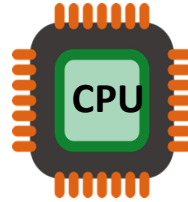
- ✧ 40+ virtual machines



- ✧ 14+ PBs of Storage



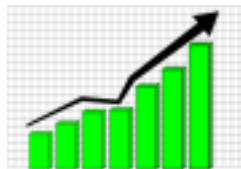
- ✧ 20+ TB of Memory



- ✧ 1500+ physical cores

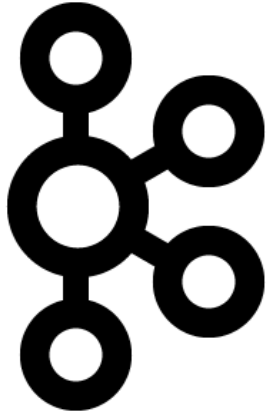


- ✧ HDDs and SSDs

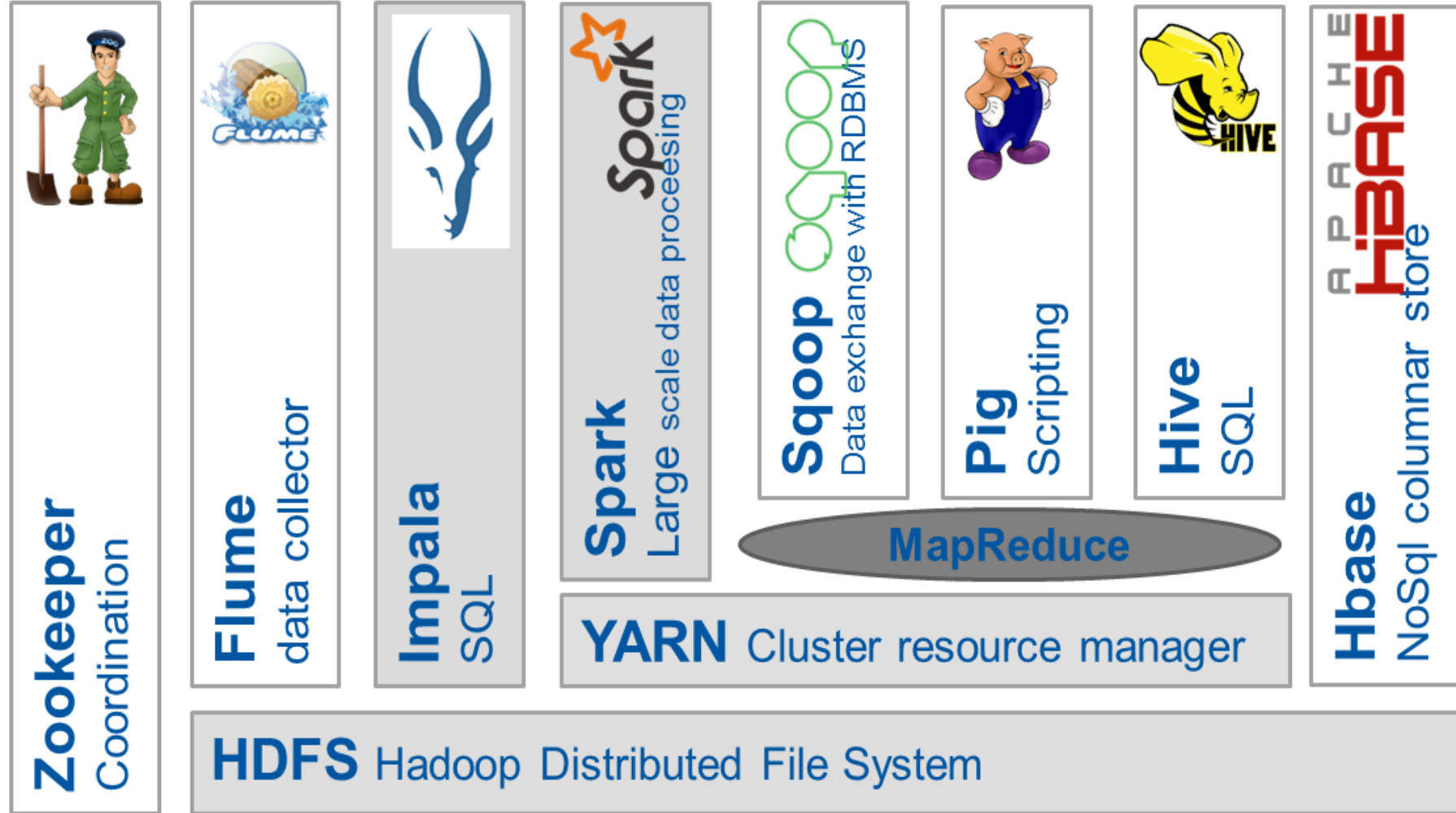


- ✧ Data growth: 4 TB per day

Overview of available components in 2018



Kafka:
streaming and
ingestion




Spark and Hadoop clusters

- Software – mixture of CERN Apache Hadoop and Cloudera Distribution for Hadoop
- Plans to align the software of all the clusters to CERN Apache Hadoop by end of 2018

Cluster Name	Configuration	Software Version	Primary Usage
nxcals	20 nodes (Cores 480, Mem - 8 TB, Storage – 5 PB, 96GB in SSD)	hadoop_cern_2.7.5_v1.0	Accelerator logging
analytix	48 nodes (Cores – 892,Mem – 7.5TB,Storage – 6 PB)	cdh5.7.5	General Purpose
hadalytic	14 nodes (Cores – 196,Mem – 768GB,Storage – 2.15 PB)	cdh5.7.5	Development cluster
lxhadoop	18 nodes (Cores – 288,Mem – 912GB,Storage – 1.29 PB)	cdh5.7.5	ATLAS Event Index
QA	10 nodes	hadoop_cern_2.7.5_v1.0	QA cluster

Current usage of LCG Releases and CVMFS

- From 2016, when we started the collaboration on SWAN 
- SWAN session (container) to act as Hadoop and Spark client
 - How to distribute the Hadoop and Spark binaries? Answer: LCG Releases
 - How to distribute the configuration? Answer: CVMFS
- Software concerning Hadoop and Spark Service
 - Apache Hadoop 2.7.3 available in LCG releases
 - Spark 1.6 to Spark 2.3 versions across LCG releases
- Spark and Hadoop cluster configuration files on CVMFS
 - HADOOP CONFIG: `/cvmfs/sft.cern.ch/lcg/etc`

Current usage of LCG Releases - continued

- SWAN

- The client binaries for Spark and Hadoop are sourced from LCG Releases
- The cluster configuration for Spark and Hadoop are sourced from CVMFS
- The runtime environment of Spark application components – master (SWAN) and workers (Spark cluster nodes) are configured exactly in the same way, sourcing software from LCG Releases

- Hadoop-Client (bare-metal / VM)

- Mount CVMFS
- Source software and configuration from LCG Releases

```
class{'::cvmfs':  
  mount_method => 'mount',  
}  
cvmfs::mount { 'sft.cern.ch':  
  require =>  
  Cvmfs::Domain['cern.ch'],  
}
```

- `source /cvmfs/sft.cern.ch/lcg/views/LCG_93/x86_64-centos7-gcc62-opt/setup.sh`
- `source /cvmfs/sft.cern.ch/lcg/etc/hadoop-confext/hadoop-setconf.sh analytix`

- Recommended Spark and Hadoop client setup by the service

Current usage of LCG Releases - continued

- Access to Hadoop and Spark service from lxplus
 - CERN user with authorization to Hadoop and Spark service can run spark jobs and hdfs commands thanks to the software in LCG Releases and the configuration in CVMFS
 - KB: <https://cern.service-now.com/service-portal/article.do?n=KB0004426>
- Users of PySpark (python on spark) profit from the huge number of packages available in python distribution. The service offloads maintenance of python distribution to LCG Releases
- Hadoop and Spark service relies on LCG Releases to deliver stable Hadoop-client and configuration to our user community
- LCG Releases are made available on the service nodes by 1) Installing *HEP_Oslibs* and 2) mounting *sft.cern.ch* using CVMFS

Future Requirements

- Deployment of CERN Apache Hadoop
 - Essentially it is the upstream version with required fixes backported from later versions
 - repo - https://:@gitlab.cern.ch:8443/db/hadoop_dist/hadoop/cerndb-sw-hadoop-source.git
 - tag - `hadoop_cern_2.7.5_v1.0`
- Apache Spark is actively being developed and new versions are released at **short timescales**, which most of our users would like to use, so a **stable working bleeding edge** would be very nice
- Security fixes and bug fixes to software stay in bleeding edge stack for long time due to longer publication dates, again **stable working bleeding edge** would be very nice

Future Requirements

- Multiple versions of the same component (e.g Apache Spark) in a LCG release?
- Automate the deployment of new releases into bleeding edge?
Requested by our clients – NXCals project (BE-CO)