TOWARDS PHYSICS DATA ANALYSIS AND DATA REDUCTION WITH APACHE SPARK

E. Motesnitsalis[1], L. Canali[1], V. Khristenko[1], M. Cremonesi[2], J. Pivarski[3], O. Gutsche[2]

[1)]CERN, Geneva, Switzerland [2)]Fermilab, Batavia, USA [3)]Princeton University

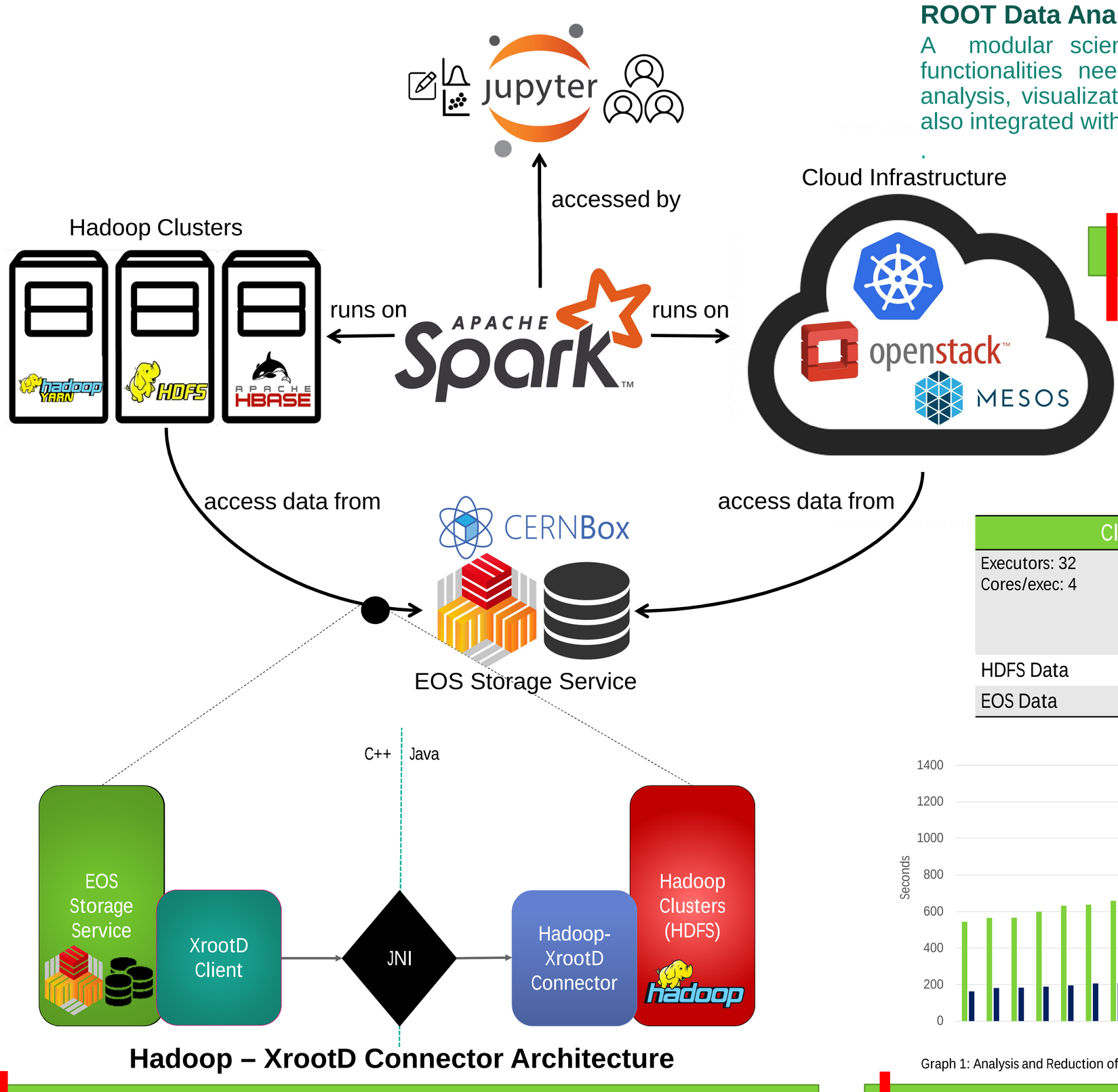
GOAL AND MOTIVATION

- The goal of the project is to perform Physics Data Analytics and Data Reduction with Big Data Technologies over data acquired from CMS, one of the four big detector of the Large Hadron Collider (LHC) at CERN in Geneva, Switzerland
- We are interested in investigating new ways to analyse physics data and improve resource utilization and time-to-physics
- We want to adopt new technologies widely used in the industry
- In this way, we open the High Energy Physics (HEP) field to a larger community of data scientists and we improve the chances of researchers on the job market outside academia

EOS Service

A disk-based, low-latency storage service with a highly-scalable hierarchical namespace, which enables data access through the XROOTD protocol. It provides storage for both physics and user use cases via different service instances such as EOSPUBLIC, EOSATLAS, EOSCMS.

Planned Architecture Overview



Hadoop – XrootD Connector Architecture

CURRENT PROCEDURES AND PROGRESS TO DATE

- Until today, the vast majority of high energy physics analysis is done with the ROOT Framework which uses physics data that are stored in ROOT format files.
- CERN uses the EOS storage service to serve files to the ROOT framework
- To work with Physics Analysis on Spark, we have solved two main challenges:
  1. Read files in ROOT Format using Spark
  2. Access files stored in EOS directly from Hadoop/Spark
- We now have fully functioning Analysis & Reduction examples tested over CMS Open Data (1 TB)

SPARK-ROOT ARCHITECTURE DETAILS

- Spark-root is a SCALA library developed by DIANA-HEP which implements DataSource for Apache Spark like Parquet/AVRO
- It can read ROOT TTrees, infer their schema and import ROOT files to Spark Dataframes/Datasets/RDDs

ROOT Data Analysis Framework

A modular scientific software framework which provides all the functionalities needed to deal with big data processing, statistical analysis, visualization and file storage. It is mainly written in C++ but also integrated with Python and R.

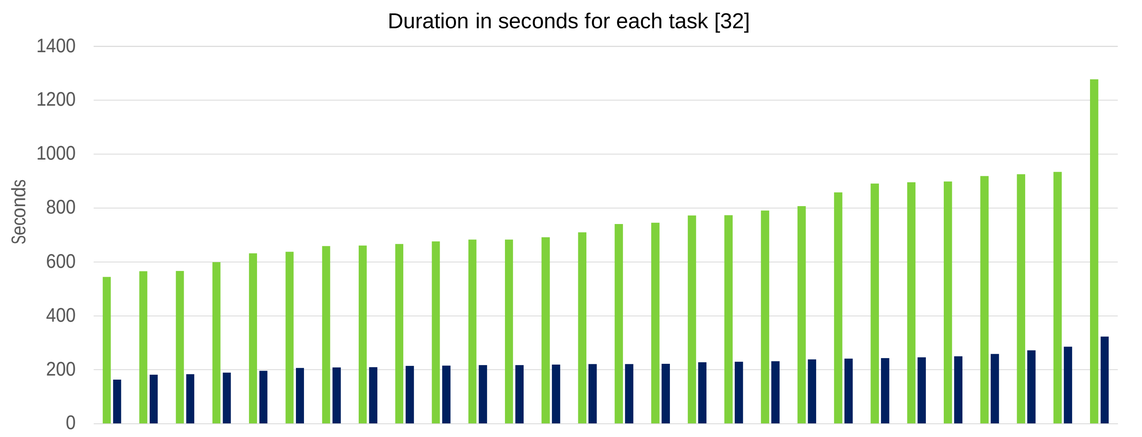
.

ROOT Data Analysis Framework

HADOOP-XROOTD ARCHITECTURE DETAILS

- Hadoop-XrootD Connector is a JAVA library that connects to the XrootD client via JNI
  - It reads files from the EOS Service directly, without the need to import/export files to HDFS
- Currently it is working with 30% the speed of direct HDFS access for a READ operation

Cluster Testing – Data Analysis and Reduction					
Executors: 32 Cores/exec: 4	Data Input	Total Executor Time	Sum of task CPU time	Read time as elapsed time – CPU time	Run time
HDFS Data	0.5 TB	5.8h	2.9h	2.9h	5 min
EOS Data	0.5 TB	11.7h	2.9h	8.8h	19 min



Graph 1: Analysis and Reduction of 0.5 TB of CMS Open Data stored in EOS and HDFS respectively

FUTURE STEPS

- Hadoop-XrootD Connector:
  - Introduce optimizations and tuning to decrease the performance gap between EOS and HDFS
- Spark-root:
  - Implement the ability to write to ROOT files
- Investigate scaling behaviour for larger input -starting with 10 TB / 100 TB and aiming to reach 1 PB- with the help of Intel® CoFluent™ Technology for Big Data simulation based cluster level optimization
- Investigate new ways to deploy Spark over Openstack with Apache Mesos and Kubernetes

SUMMARY AND OUTLOOK

- We have solved two important challenges for performing Physics Analysis over Spark:
  - Hadoop-XrootD Connector can directly access files from the EOS Service
  - Spark-root can read ROOT files and infer their schema into Spark
- Our long term goal is to provide a unified service accessible via Notebook Software such as Jupyter Notebook that will enable users to run Spark in multiples ways, including Bare-Metal Hadoop Clusters or virtualized Openstack Cloud Infrastructure with Kubernetes or Apache Mesos
- This will simplify the existing physics data analysis procedures by improving resource utilization, providing user-friendly environments, and reducing the overall “time-to-physics”