SWAN and Spark on Kubernetes discussion

IT-DB-SAS, 10th Oct 2018
Prasanth Kothuri

Hosted notebook service

- Why?
  - Interactive data analysis
  - Data exploration
  - Prototyping ETL, ML workflows
  - Unified (integrations with analysis ecosystems)
  - Reduce the complexity of working with distributed systems

- What?
  - use and share notebooks with others without having to download, install, or run anything on your own computer other than a browser
  - Integrations with CERN core services (e.g. SSO, ldap, egroups)
  - Storage to store notebooks and share notebooks
  - Software: HEP packages and widely used analysis ecosystems (python, R)

- Who?
  - NxCALS
  - WLCG and IT Monitoring
  - BE Industrial Controls
  - Experiments (depending on ROOT RDataFrame)

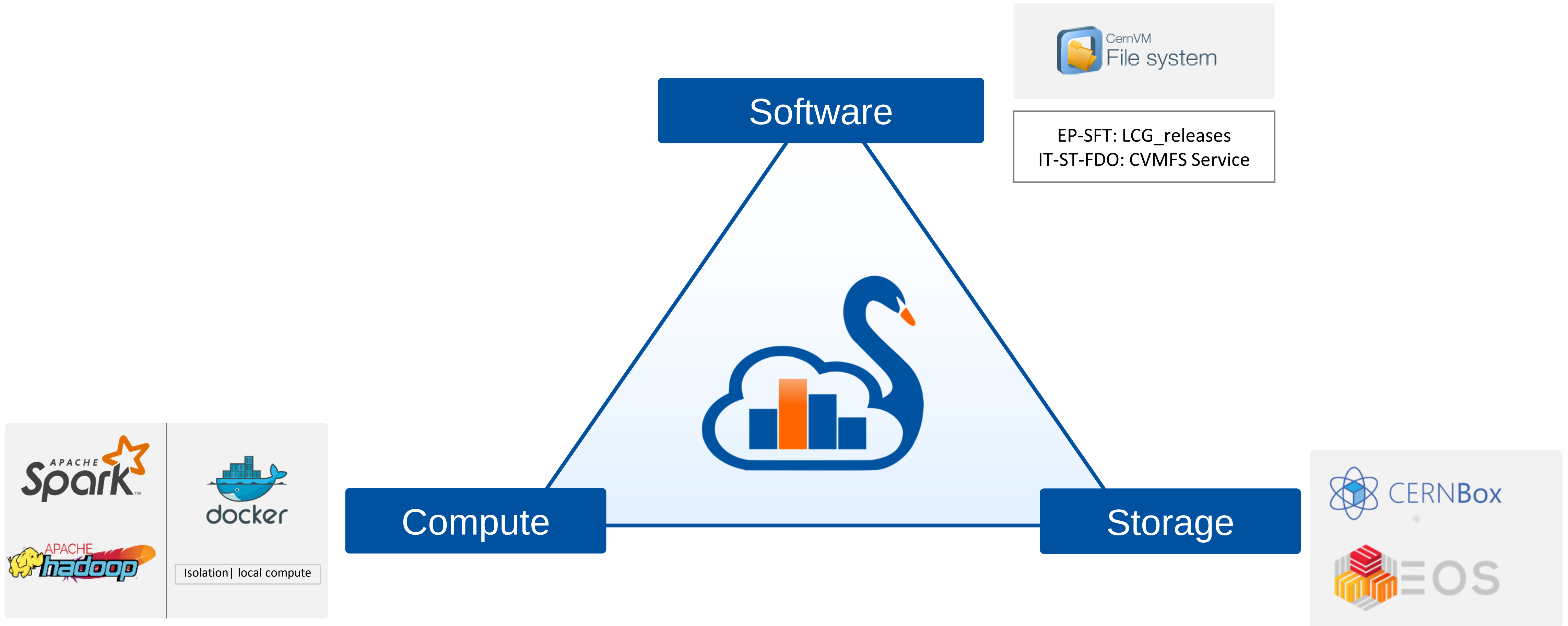
SWAN – Introduction

- SWAN – Service for Web based ANalysis
  - collaboration between EP-SFT, IT-ST and IT-DB

- Analysis from a web browser
  - Integrated with other analysis ecosystems: ROOT C++, Python and R
  - Ideal for exploration, reproducibility, collaboration
  - Available everywhere and at any time

- Integrated with CERN services [1]
  - Software: CVMFS
  - Storage: CERNBox, EOS
  - Compute: local (docker)
  - Scalable Analytics: Fully Integrated with IT Spark and Hadoop Clusters
    - powerful and scalable platform for data analysis
    - Python on Spark (PySpark) at scale

SWAN - Integrating Services



EP-SFT: LCG_releases
IT-ST-FDO: CVMFS Service

IT-DB-SAS
Hadoop Service

IT-ST-FDO
EOS Service

[1] SWAN team consists of members from EP-SFT, IT-DB and IT-ST groups

SWAN – Jupyter notebooks on demand

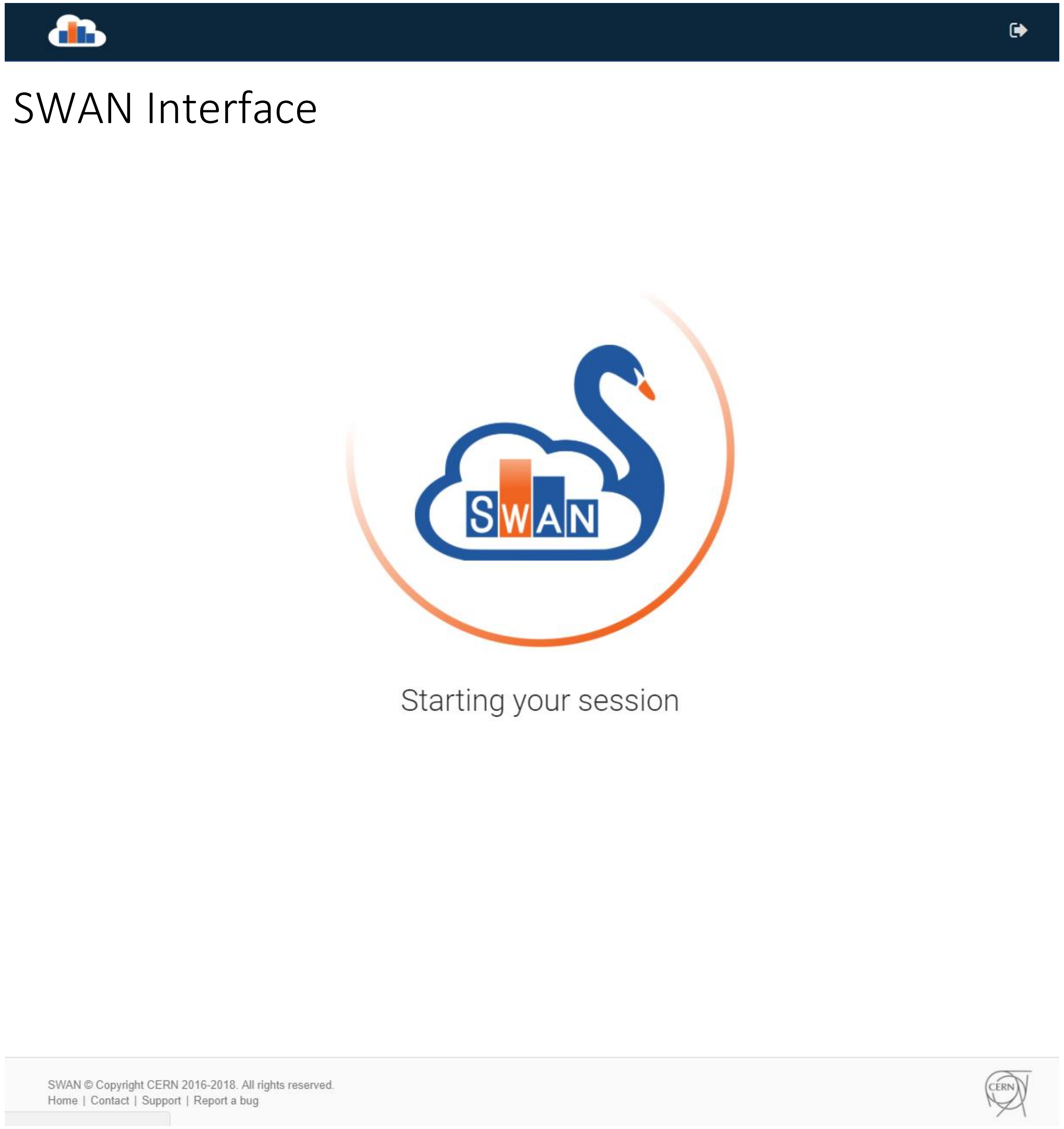
- A web-based interactive interface and platform that combines code, equations, text and visualisations

- Many supported languages (kernels)
  - In SWAN: Python, ROOT C++, R and Octave

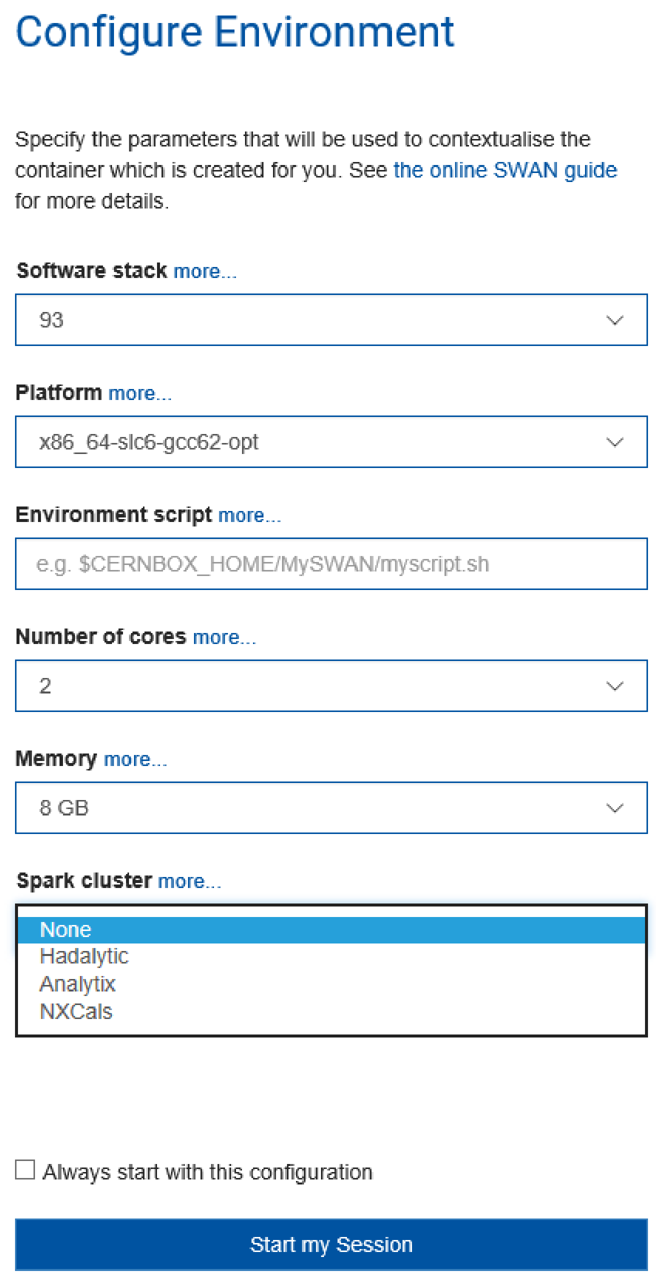
- Interactive, usually lightweight computations and now distributed parallel processing capability with the integration of mass processing system (Apache Spark)

- Very useful for multiple use cases
  - Analysis, Exploration, Teaching, Documentation and Reproducibility

SWAN Interface



SWAN © Copyright CERN 2016-2018. All rights reserved.
Home | Contact | Support | Report a bug



Configure Environment

Specify the parameters that will be used to contextualise the container which is created for you. See the online SWAN guide for more details.

Software stack more...
93

Platform more...
x86_64-slc6-gcc62-opt

Environment script more...
e.g. $CERNBOX_HOME/MySWAN/myscript.sh

Number of cores more...
2

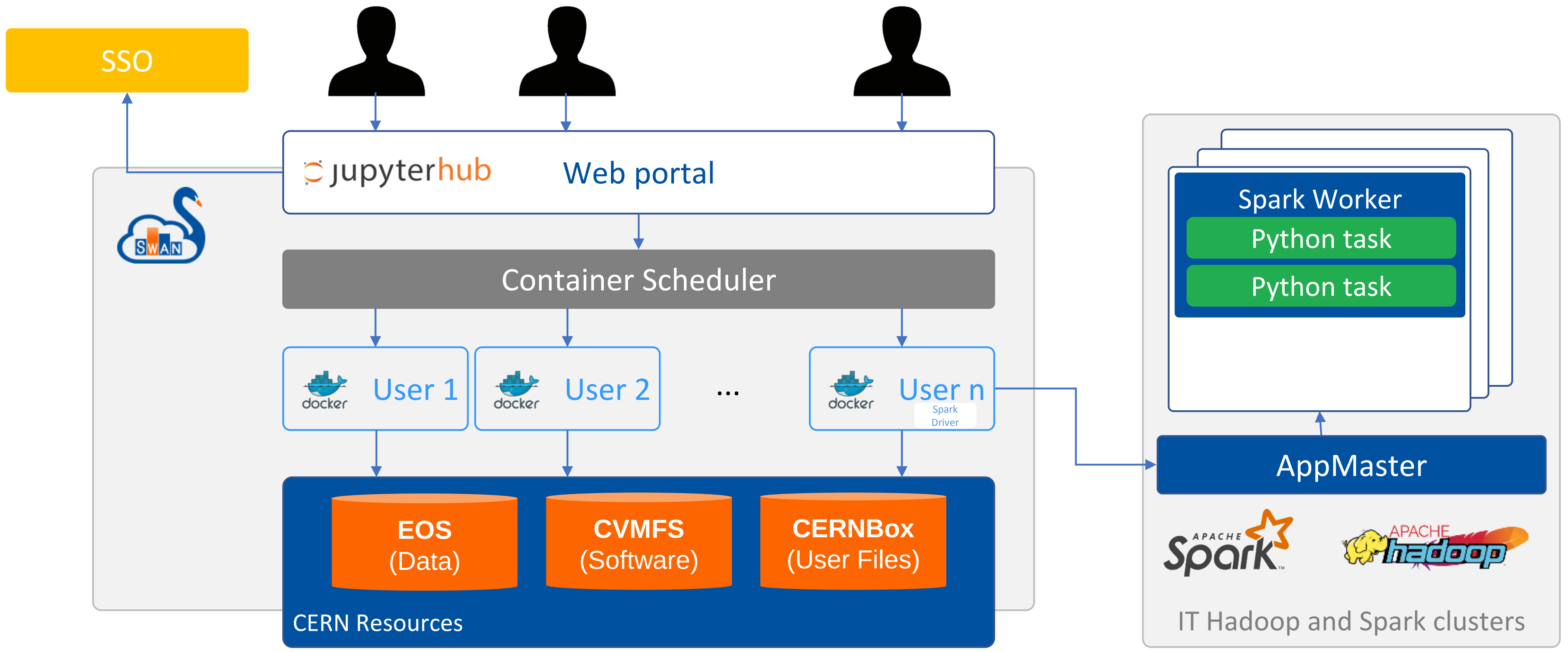
Memory more...
8 GB

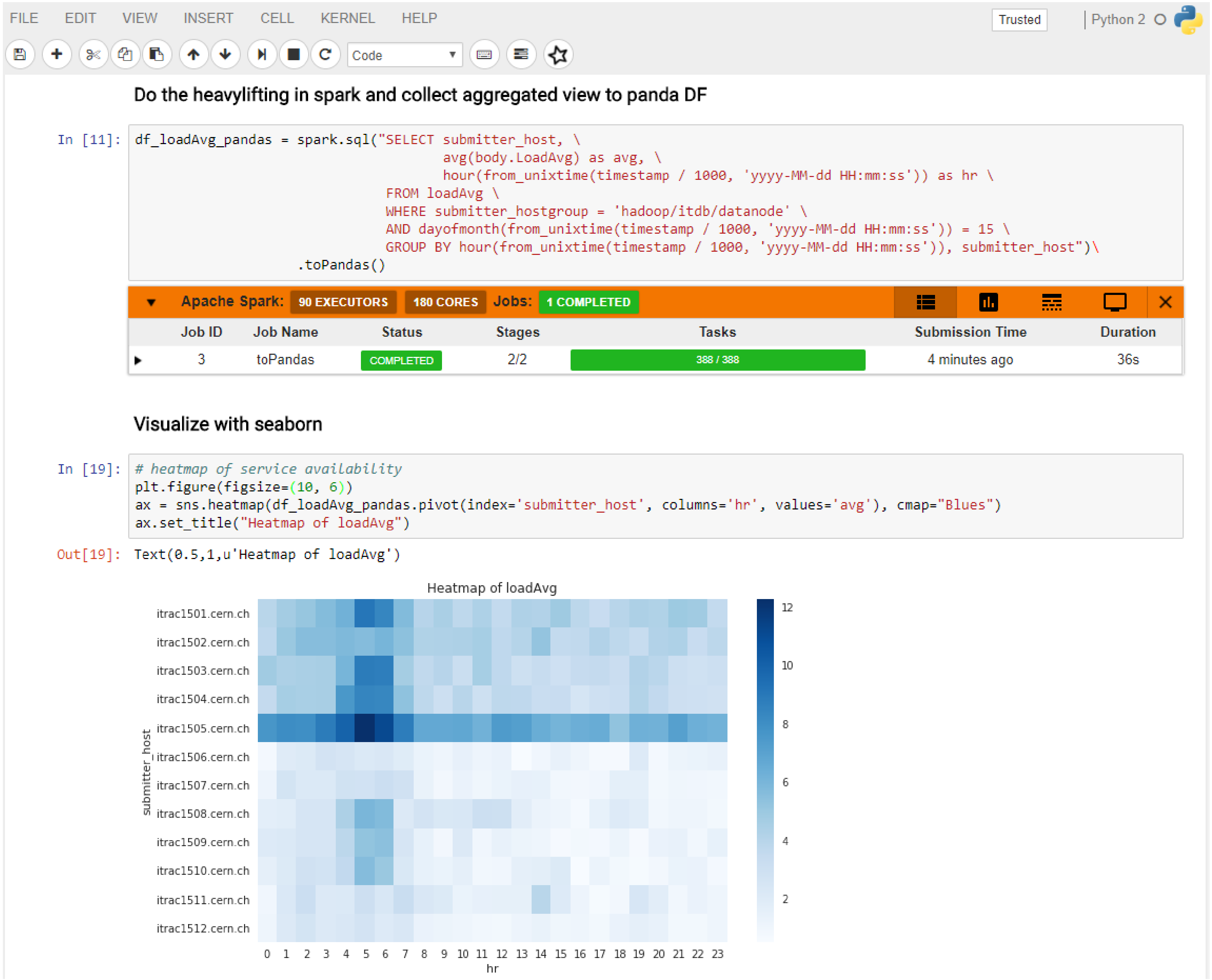
Spark cluster more...
None
Hadalytic
Analytix
NXCals

☐ Always start with this configuration

Start my Session

SWAN – Architecture





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()
```

Apache Spark: 90 EXECUTORS 180 CORES Jobs: 1 COMPLETED

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

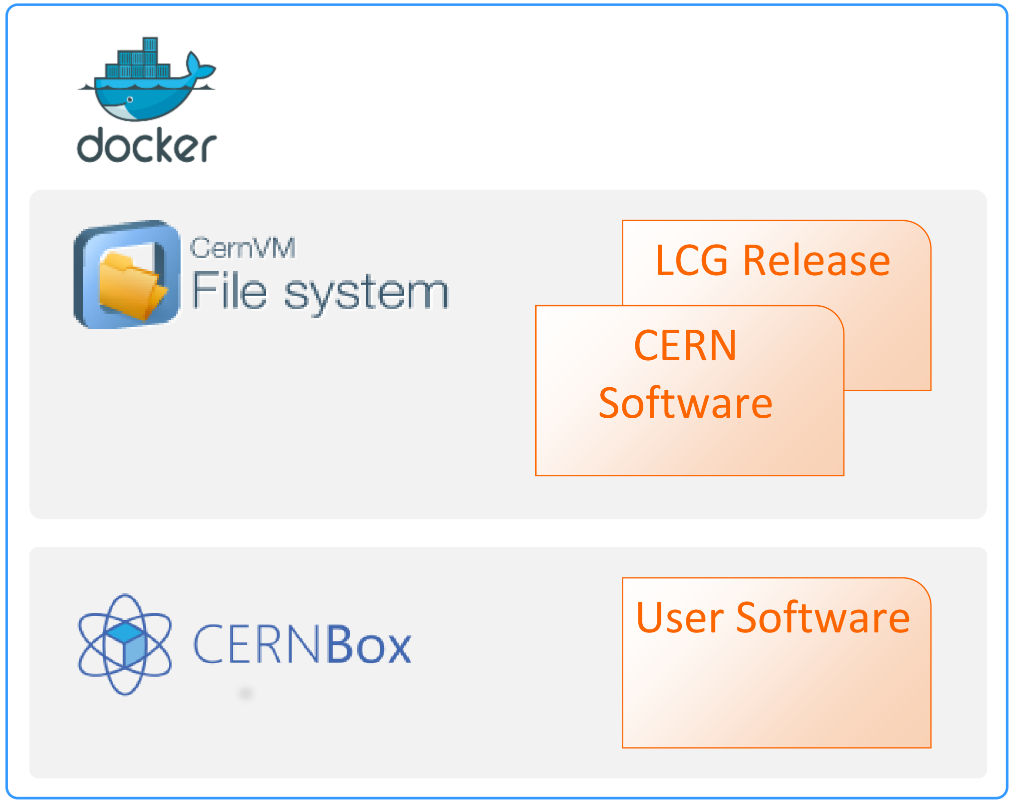
Code

Monitoring

Visualizations

Software - CVMFS

- Docker: single thin image, managed by the service
- CVMFS: delivery of experiments and beams software
  - ”LCG Releases”[1] – hundreds of packages coherently built
  - Software used by researchers is available
- CERNBox: possibility to further customize user environment by installing additional libraries in user local storage



[1] http://lcginfo.cern.ch

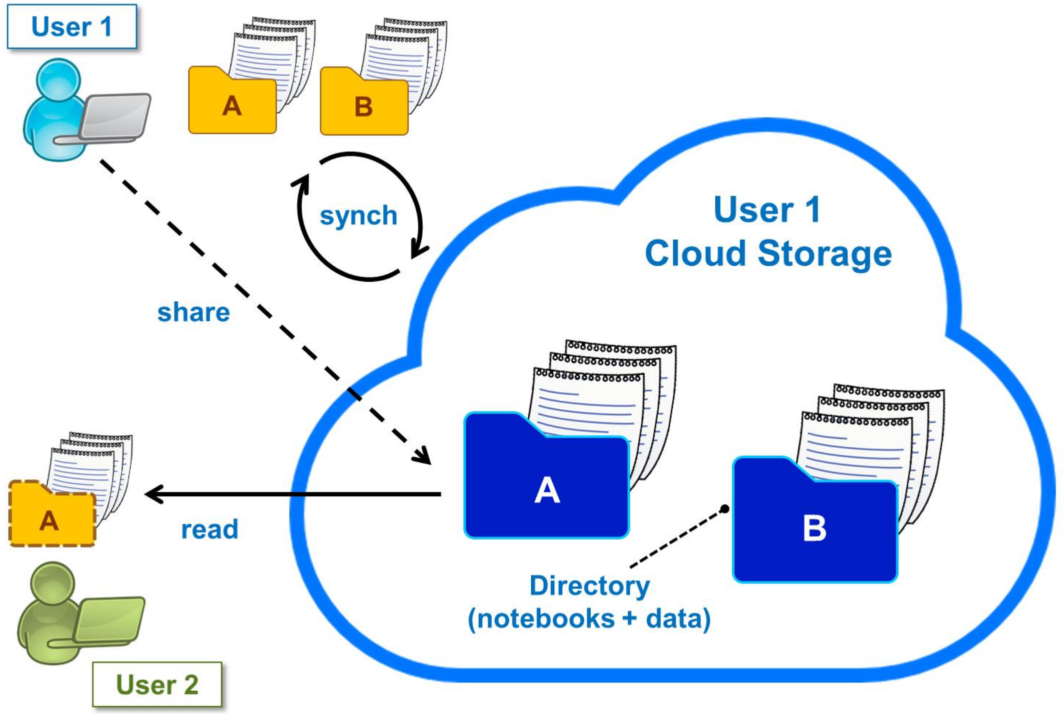
Storage - EOS

- Uses EOS mass storage system

  All experiment data potentially available

- User personal space, synchronized through CERNBox

  All files synced across devices, the cloud and other users



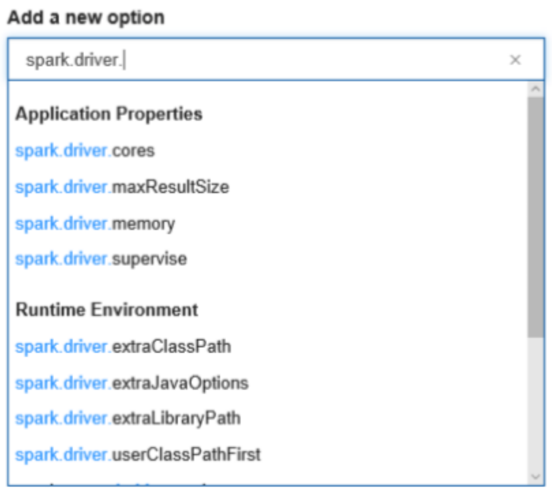
Scalable Analytics: Spark-clusters with SWAN integration

- Apache Spark is a highly scalable, unified analytics engine for large-scale data processing
- Built for complex analytics, streaming analytics and machine learning
- Usage of Apache Spark is growing at CERN

Cluster Name	Configuration	Primary Usage	
nxcals	20 nodes (Cores 480, Mem - 8 TB, Storage – 5 PB, 96GB in SSD)	Accelerator logging (NXCALS) project dedicated cluster	✓
analytix	48 nodes (Cores – 892,Mem – 7.5TB,Storage – 6 PB)	General Purpose	✓
hadalytic	14 nodes (Cores – 196,Mem – 768GB,Storage – 2.15 PB)	Development cluster	✓

SWAN_Spark features

- Spark Connector – handling the spark configuration complexity
  - User is presented with Spark Session (Spark) and Spark Context (sc)
  - Ability to bundle configurations specific to user communities
  - Ability to specify additional configuration



Add a new option

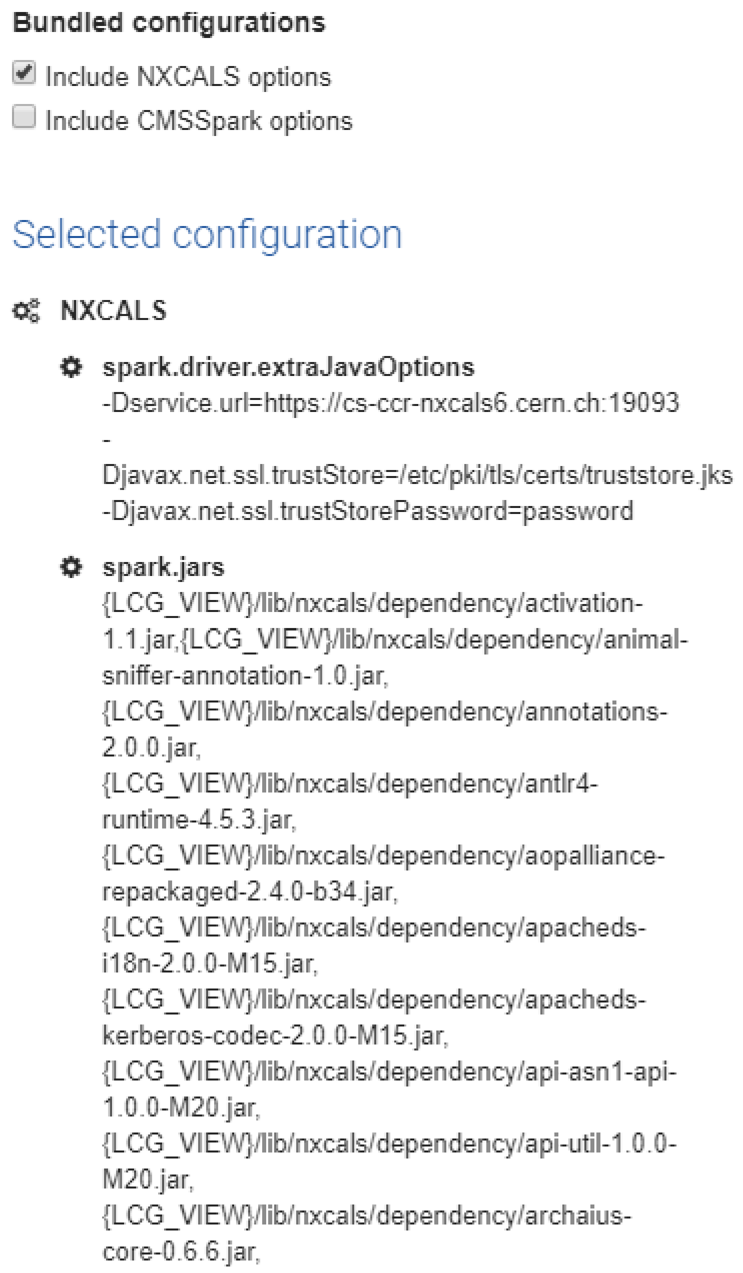
spark.driver.

Application Properties

spark.driver.cores
spark.driver.maxResultSize
spark.driver.memory
spark.driver.supervise

Runtime Environment

spark.driver.extraClassPath
spark.driver.extraJavaOptions
spark.driver.extraLibraryPath
spark.driver.userClassPathFirst



Bundled configurations

[✓] Include NXCALS options
[ ] Include CMSSpark options

Selected configuration

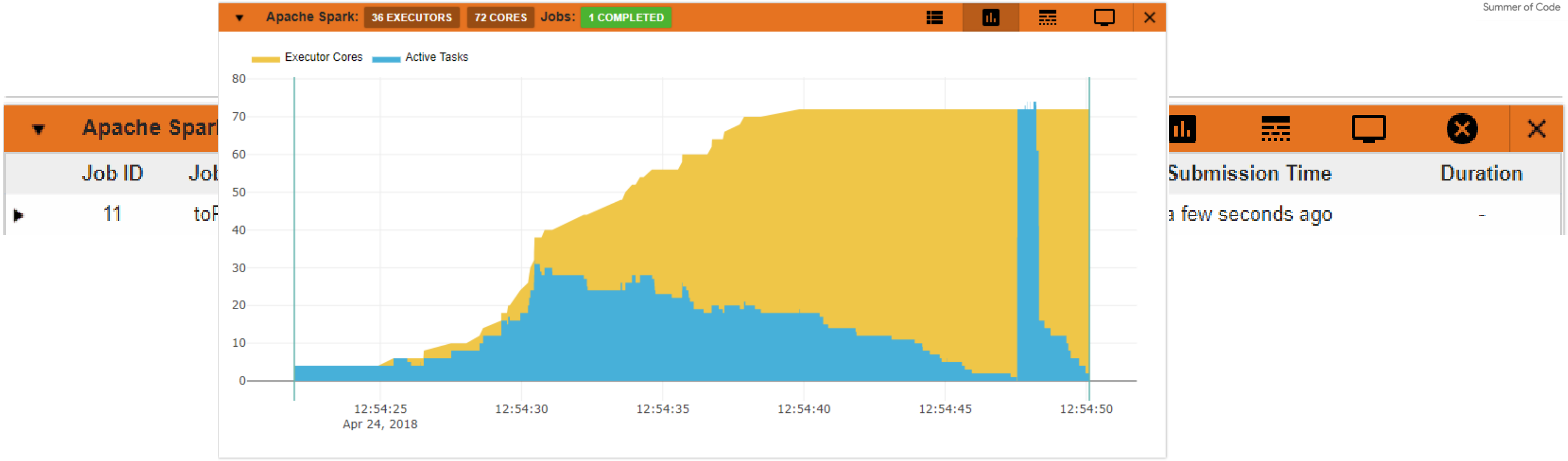
NXCALS

spark.driver.extraJavaOptions
-Dservice.url=https://cs-ccr-nxcals6.cern.ch:19093 -Djavax.net.ssl.trustStore=/etc/pki/tls/certs/truststore.jks -Djavax.net.ssl.trustStorePassword=password

spark.jars
{LCG_VIEW}/lib/nxcals/dependency/activation-1.1.jar,{LCG_VIEW}/lib/nxcals/dependency/animal-sniffer-annotation-1.0.jar, {LCG_VIEW}/lib/nxcals/dependency/annotations-2.0.0.jar, {LCG_VIEW}/lib/nxcals/dependency/antlr4-runtime-4.5.3.jar, {LCG_VIEW}/lib/nxcals/dependency/aopalliance-repackaged-2.4.0-b34.jar, {LCG_VIEW}/lib/nxcals/dependency/apacheds-i18n-2.0.0-M15.jar, {LCG_VIEW}/lib/nxcals/dependency/apacheds-kerberos-codec-2.0.0-M15.jar, {LCG_VIEW}/lib/nxcals/dependency/api-asn1-api-1.0.0-M20.jar, {LCG_VIEW}/lib/nxcals/dependency/api-util-1.0.0-M20.jar, {LCG_VIEW}/lib/nxcals/dependency/archaius-core-0.6.6.jar,

SWAN_Spark features

- Spark Monitor – jupyter notebook extension
  - For live monitoring of spark jobs spawned from the notebook
  - Access to Spark WEB UI from the notebook
  - Several other features to debug and troubleshoot Spark application
  - Developed in the context of HSF Google Summer of Code program [1]



[1] http://hepsoftwarefoundation.org/gsoc/2017/proposal_ROOTspark.html

Authentication and Encryption

- Authentication
  - spark.authenticate : authentication via shared secret, ensures that all the actors (driver, executor, AppMaster) share the same secret

- Encryption
  - encryption is enabled for all spark application services (block transfer, RPC etc)

- Further details on SWAN_Spark security model
  - https://cernbox.cern.ch/index.php/s/B4IdwuuhJ0TWgtH

Industry focus – Unified Big Data analytics platforms

databricks

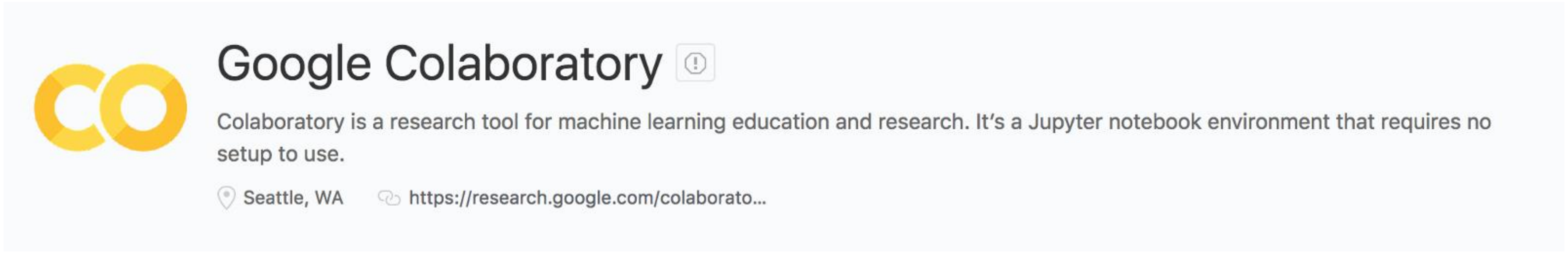
Databricks Unified Platform

- Simplifying Big Data and AI

cloudera

Cloudera Data Science Workbench

- Enables fast, easy and secure self-service data science



Google Colaboratory

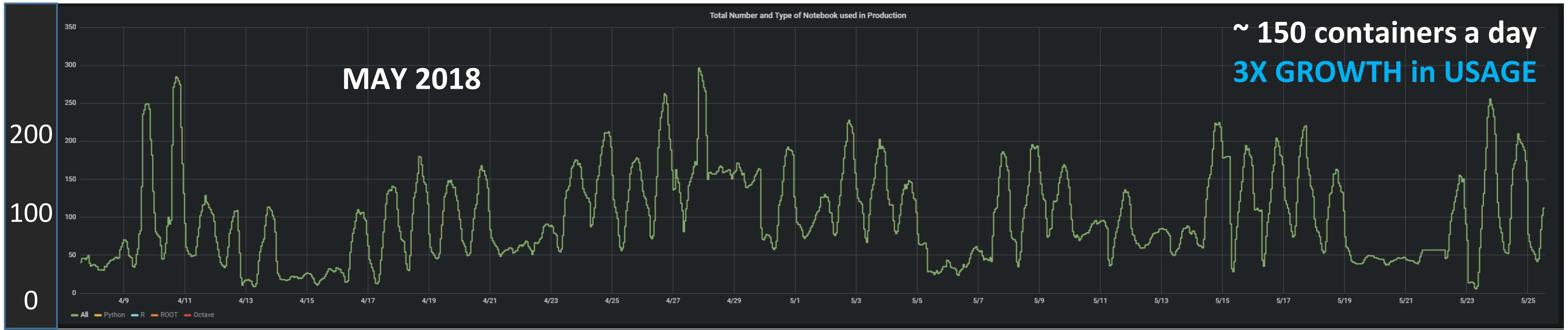
Colaboratory is a research tool for machine learning education and research. It's a Jupyter notebook environment that requires no setup to use.

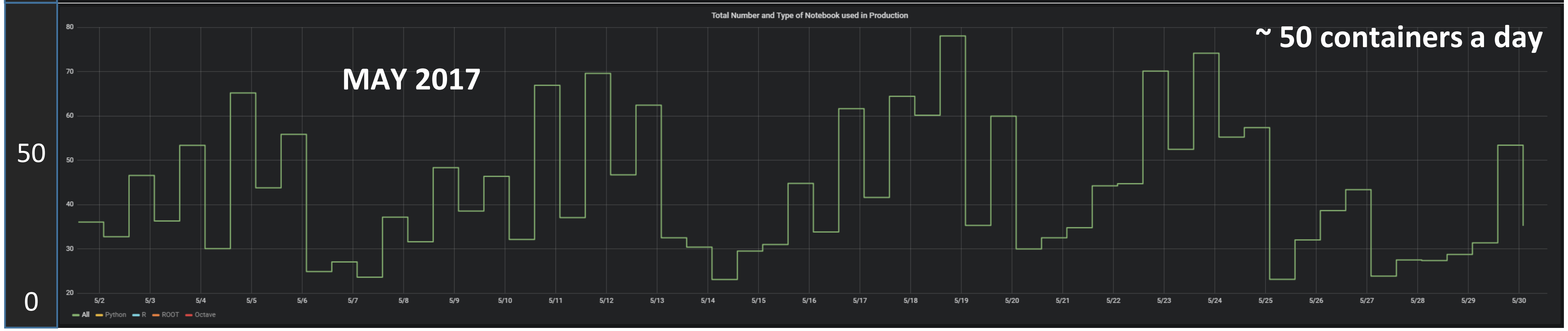
Seattle, WA https://research.google.com/colaborato...

SWAN

<u>Comparable to industry offerings with integrations for CERN / HEP data and compute</u>

Growing usage and reliance on SWAN





Further growth in usage expected with the integration of SPARK clusters and onboarding of BE NXCALS users

Contribution from IT-DB-SAS

Development of Spark Connector

Development of Spark Monitor under GSoC project

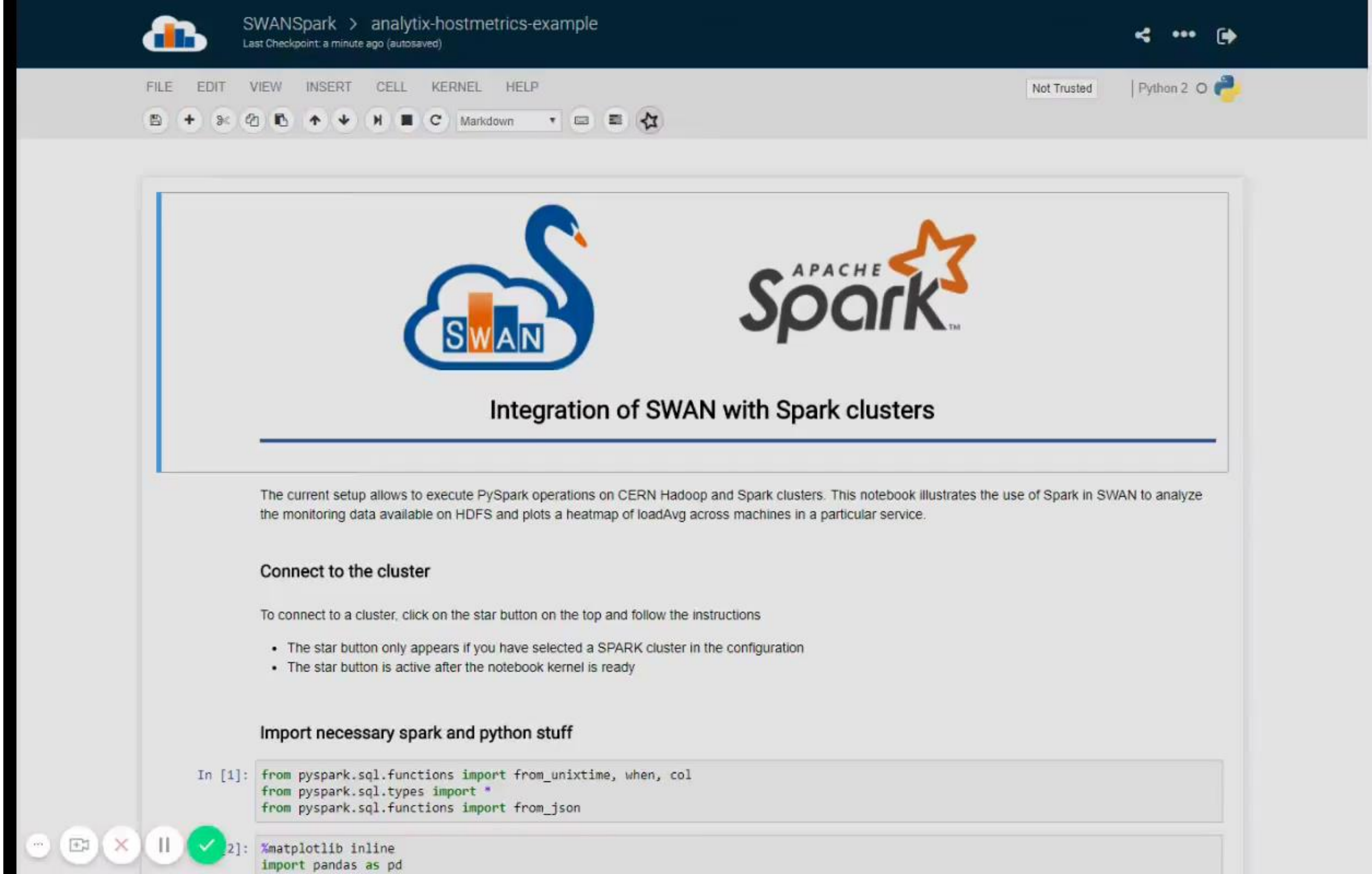
Development of solution of publish hadoop/spark configuration to CVMFS

Development of hdfsBrowser jupyter extension

Publishing of software to CVMFS

Supporting NXCals team to adapt SWAN solution

SWAN_Spark – Demo


SWANSpark > analytix-hostmetrics-example
Last Checkpoint: a minute ago (autosaved)
FILE EDIT VIEW INSERT CELL KERNEL HELP
Not Trusted
Python 2
Markdown

Integration of SWAN with Spark clusters

The current setup allows to execute PySpark operations on CERN Hadoop and Spark clusters. This notebook illustrates the use of Spark in SWAN to analyze the monitoring data available on HDFS and plots a heatmap of loadAvg across machines in a particular service.

Connect to the cluster

To connect to a cluster, click on the star button on the top and follow the instructions

- The star button only appears if you have selected a SPARK cluster in the configuration
- The star button is active after the notebook kernel is ready

Import necessary spark and python stuff

```
In [1]: from pyspark.sql.functions import from_unixtime, when, col
        from pyspark.sql.types import *
        from pyspark.sql.functions import from_json
```

```
    2]: %matplotlib inline
        import pandas as pd
```

Future work and outlook

Ability to spawn and attach to disposable containerized Spark clusters

Improving the authentication mechanism to access spark clusters

- Avoids (double) typing of password to access spark clusters

HDFS browser & Datasets

- ability to browse HDFS from SWAN
- abstraction to create and share datasets

Job submission to Spark clusters

- SWAN user session is a full-fledged Hadoop-Spark client

Support and evolution of Spark aspects of the service

Takeover of SWAN service as it better fits the mandate of IT-DB-SAS ?

Moving Forward

Continue collaboration on SWAN Service with the following improvements

- Build the knowledge and documentation on SWAN service
- Open the service to allow support and contributions from IT-DB-SAS

Run a separate instance of SWAN of NxCALS

- Gives a good starting point
- Evolve based on big data / distributed computing needs

Develop (yet another) notebook service

- possible duplication of work?

SWAN support channels

- Support ticket via SNOW (FE: SWAN), general feedback welcome to
  - swan-admins@cern.ch
  - swan-talk@cern.ch

Hadoop and Spark support channels

- Support ticket via SNOW (FE: Hadoop and Spark support), general feedback welcome to
  - ai-hadoop-admins@cern.ch

Spark on Kubernetes

Spark on Kubernetes service

- Why?
  - Physics Analysis using Spark and ML using Spark
  - Storage is external (EOS, Kafka)
  - Elasticity & Isolation
  - Cloud Native (shared environments, custom flavors)

- What?
  - Integrations with CERN infrastructure (OpenStack, Magnum)
  - Ease of job submission and management (SparkOperator)
  - Integration with Data Analysis Platform (SWAN)
  - Hadoop-XrootD connector to integrate with mainstream analysis tools
  - Integrate with physics analysis framework (ROOT RDataFrame)

- Who?
  - Physics Analysis with ROOT RDataframe
  - Spark Streaming
  - CMS Data Reduction (in future possibility ATLAS)

Investing for future!

Current Status

- Development of Spark on Kubernetes
  - Work with IT-CM Container service to discover and finalize the configuration required to deploy Spark on Kubernetes
  - Contribute to the upstream spark on kubernetes operator to add the functionality required for CERN usecases
  - Work with the users to help them productionize Spark workloads on Kubernetes
  - Contribute to the development of spark administrative guide and user guide
  - Train service managers and prospective users on Spark on Kubernetes technology

Future work and outlook

Integrate with Data Analysis platform (SWAN)

Investigate the integration of spark-on-kube interactive client mode with SWAN

Coherent monitoring of Spark workloads on Kubernetes

Develop curated examples for user communities

- Spark Streaming (for IT-CM-MM)
- TOTEM Analysis (ROOT RDataFrame)

Work with the users on adaption of new physics analysis model

Integration of Spark with HTCondor