

SWAN as interface to Hadoop clusters

Enric Tejedor for the SWAN team



<https://cern.ch/swan>

Hadoop User Forum
December 4th, 2023



Introduction



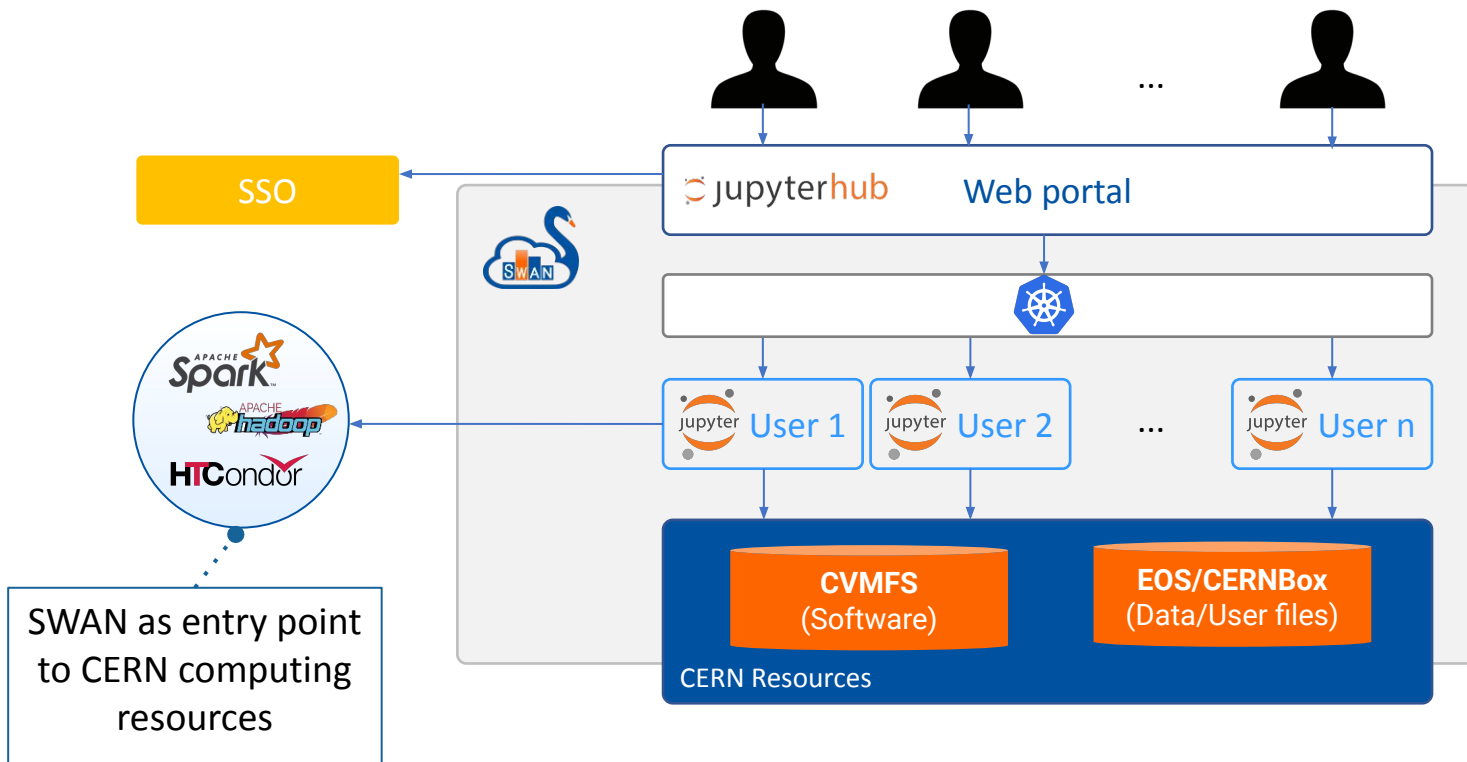


SWAN in a nutshell

- > Interactive analysis with a web browser
 - No local installation needed
 - Based on Jupyter Notebooks
 - Calculations, input data and results “in the Cloud”
- > Easy sharing of scientific results: plots, data, code
- > Good for data analysis, exploration and also teaching
- > Integration with CERN resources → added value!
 - Software (CVMFS)
 - Storage (EOS, CERNBox)
 - Computing (GPU, Hadoop / Spark, HTCondor)



SWAN architecture



SWAN-Hadoop integration





SWAN integration with Hadoop / Spark

- > SWAN is connected to Hadoop & Spark clusters at CERN
 - Physical
 - Analytix: general purpose
 - NXCALS: dedicated
 - Virtual: kubernetes cluster
- > Jupyter extensions available to:
 - Connect to a certain cluster and spawn Spark executors
 - Monitor the execution of Spark jobs
- > [Link](#) to Spark training on SWAN

Configure Environment ✕

Specify the parameters that will be used to contextualise the container which is created for you. See [SWAN service website](#) for more details and contact to administrators.

Software stack more...

Platform more...

Environment script more...

Number of cores more...

Memory more...

Spark cluster more...





Spark Connector

Spark_Simple > Spark_Simple (autosaved)

FILE EDIT VIEW INSERT CELL KERNEL WIDGETS HELP

Simple example with Spark

This notebook illustrates the use of [Spark](#) in [SWAN](#).

The current setup allows to execute [PySpark](#) operations on a local with small datasets.

In the future, SWAN users will be able to attach external Spark clusters. Moreover, a Scala Jupyter kernel will be added to use Spark from...

Import the necessary modules

The `pyspark` module is available to perform the necessary imports.

```
In [1]: from pyspark import SparkContext
```

Spark clusters connection

You are going to connect to:
analytix

You can configure the following options.
Environment variables can be used via (ENV_VAR_NAME).

Add a new option

Bundled configurations

These options will be overwritten by non-bundled options if specified

- Include CMSSpark options
- Include SparkMetrics options
- Include S3Filesystem options
- Include EOSFilesystem options
- Include PropagateUserPythonModules options
- Include UseCVMFSJavaHomeOnExecutors options
- Include ROOT_RDataFrame options

Selected configuration

SparkMetrics

- spark.cern.grafana.url
https://monit-grafana.cern.ch/d/1/sparkmetrics?orgId=23
- spark.metrics.conf.driver.sink.graphite.class

Connect

> Spark Connector – configure and establish a connection

- Uses upstream configuration provided by Hadoop team
- Provides bundled configurations
 - tailored to custom use cases
 - integrations: EOS, S3
- Allows to specify additional Spark configuration
- Connects to cluster and allocates executors
- Provides link to Spark WebUI



Spark Connector

Spark clusters connection

You are going to connect to: analytix

Spark clusters connection

You are connected to **analytix**

The following variables were instantiated

- > sc = `SparkContext`
- > spark = `SparkSession`

Connection details

- > Spark version is 3.4.1
- > Spark WebUI is available [here](#)
- > Spark Metrics Dashboard is available [here](#)
- > Spark driver logs of the running application [show/hide](#)

In [1]: `from pyspark import SparkContext, SparkSession`

> **Spark Connector** – configure and establish a connection

- Uses upstream configuration provided by Hadoop team
- Provides bundled configurations
 - tailored to custom use cases
 - integrations: EOS, S3
- Allows to specify additional Spark configuration
- Connects to cluster and allocates executors
- Provides link to Spark WebUI



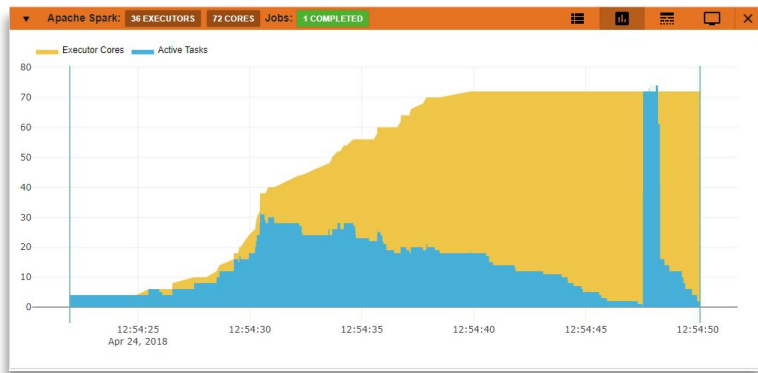
Spark Monitor

- > **Spark Monitor** – Spark monitoring from notebooks
 - For live monitoring of Spark jobs spawned from the notebook
 - Displays jobs, stages and tasks
 - A graph shows number of active tasks & executor cores vs time



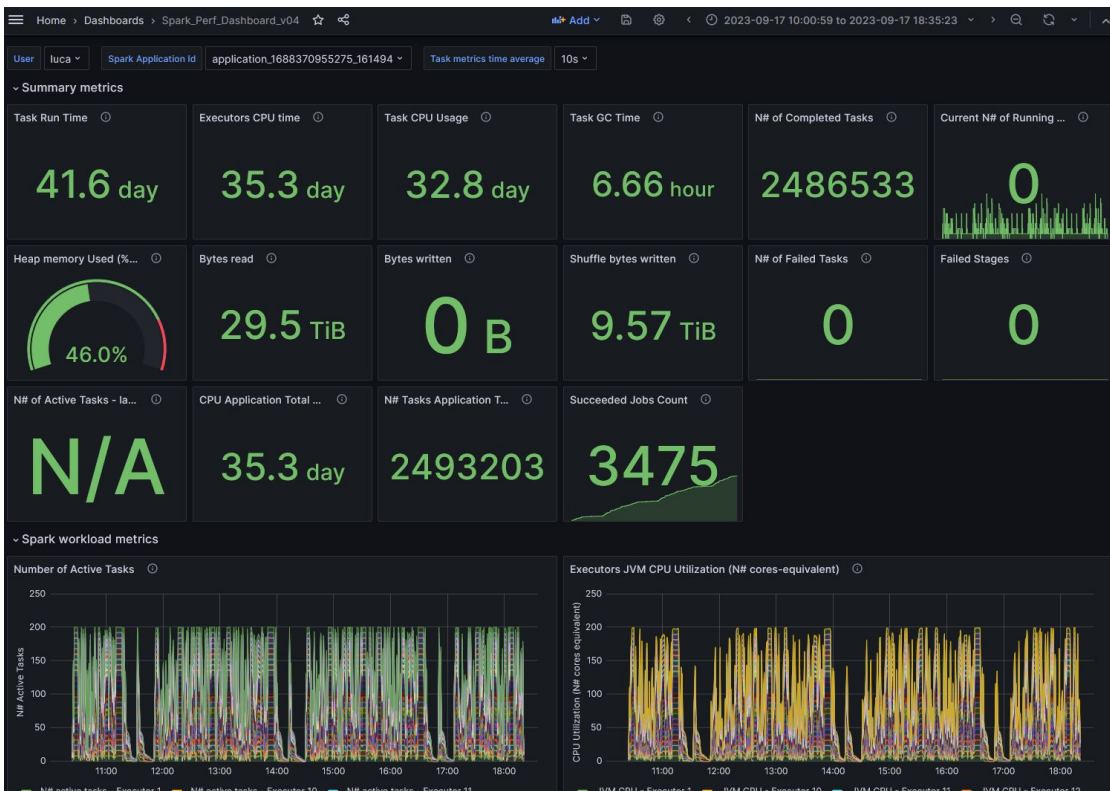
Google Summer of Code

Job ID	Job Name	Status	Stages	Tasks	Submission Time	Duration
11	toPandas	RUNNING	0/2 (1 active)	4 / 201	a few seconds ago	-





Spark Metrics



> Spark Metrics – Spark dashboard for advanced troubleshooting

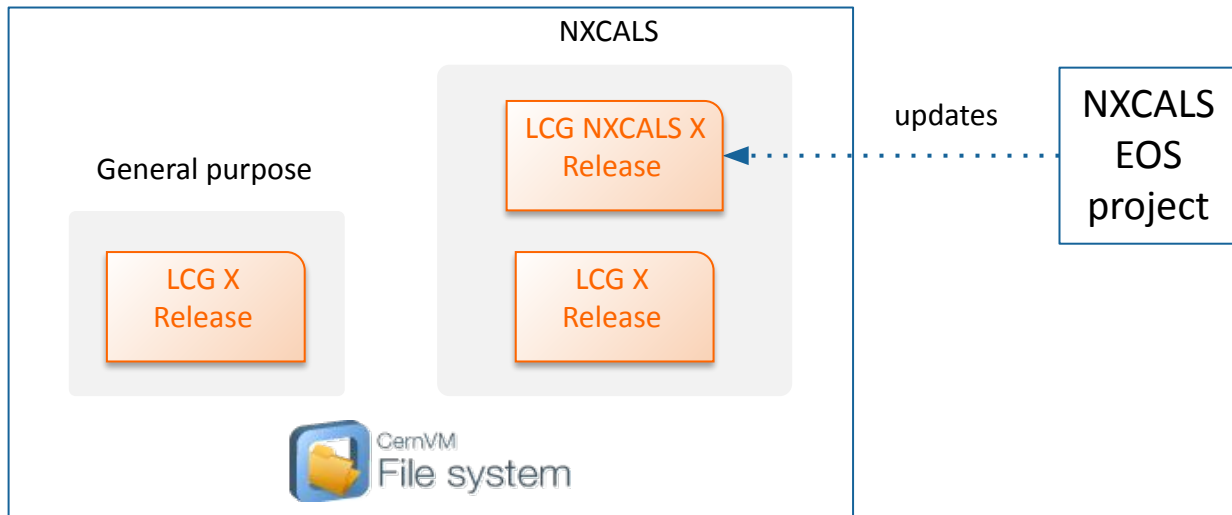
- Custom CERN development, integrating Spark instrumentation with InfluxDB and Grafana services at CERN
- Displays CPU usage, active tasks, memory, shuffle, I/O, Java GC, etc
- Real-time and historical data
- Opt-in configuration (use it when troubleshooting)





Software provisioning: CVMFS

- > Software is provisioned via curated stacks on CVMFS (LCG releases)
 - General-purpose: contain Spark, Java, Python
 - NXCALs: layered
 - Base layer: general-purpose, fixed
 - Top layer: NXCALs software, updated every day (enables faster development cycles)





Current usage

- > User session stats for SWAN kubernetes infrastructure, i.e. only **half of the user load**), last month
 - Analytix: peak of ~10
 - NXCALS: peak of ~30

Only in k8s!
~50 % of
users



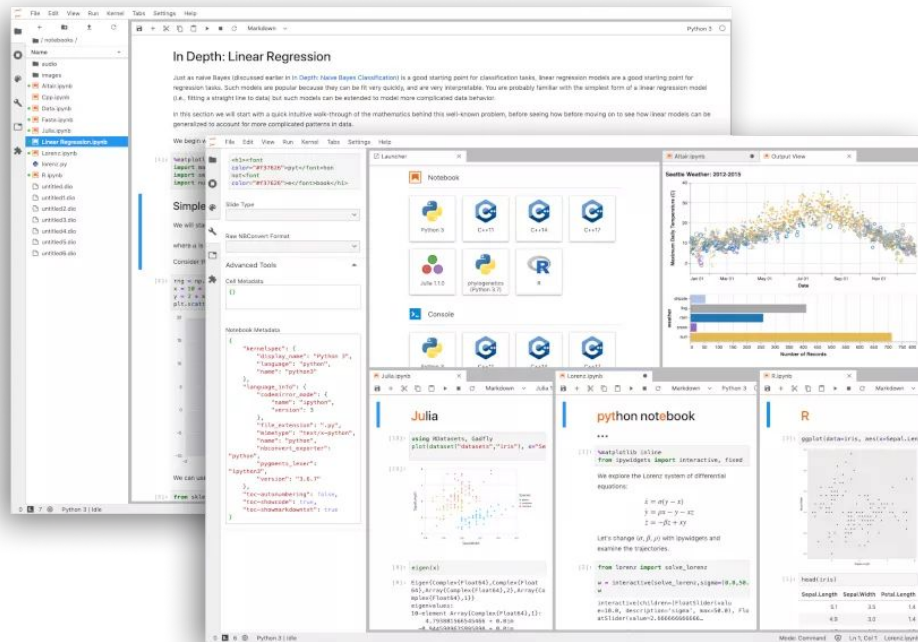
Future plans





JupyterLab

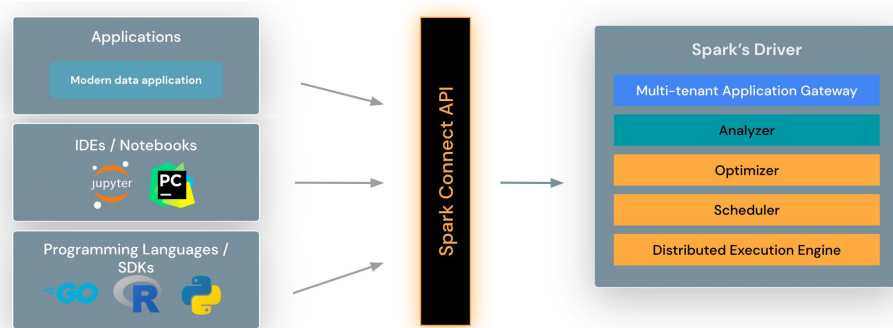
- > Latest interface proposed by project Jupyter
 - Notebooks, terminals, ...
 - ... and virtually anything via extensions
- > Will be offered as an option in test mode, before end of 2023
 - Will initially coexist with old UI
 - To be made default in the future
- > Spark extensions have been migrated to JupyterLab
 - Connector, Monitor



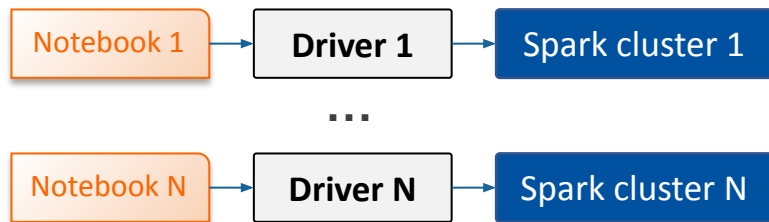


Spark Connect

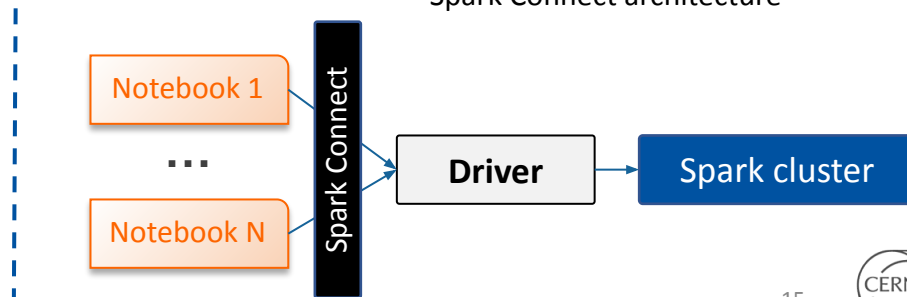
- > New client-server architecture introduced since Spark 3.4
 - Allows to spawn a server to which clients can connect
- > Under investigation in SWAN
 - Allows multiple notebooks (clients) to share the same driver (server) and executors



Classic architecture (current in SWAN)



Spark Connect architecture



Thank you

