

CERN's Platform for Data Analysis with Spark

Enric Tejedor, Prasanth Kothuri, CERN

#SAISExp1



Outline

1. What we do at CERN
2. Interactive data analysis with Spark
3. Use cases
 1. Physics data
 2. Controls data

Founded in 1954

Mission: fundamental research in Physics



Accelerating Science

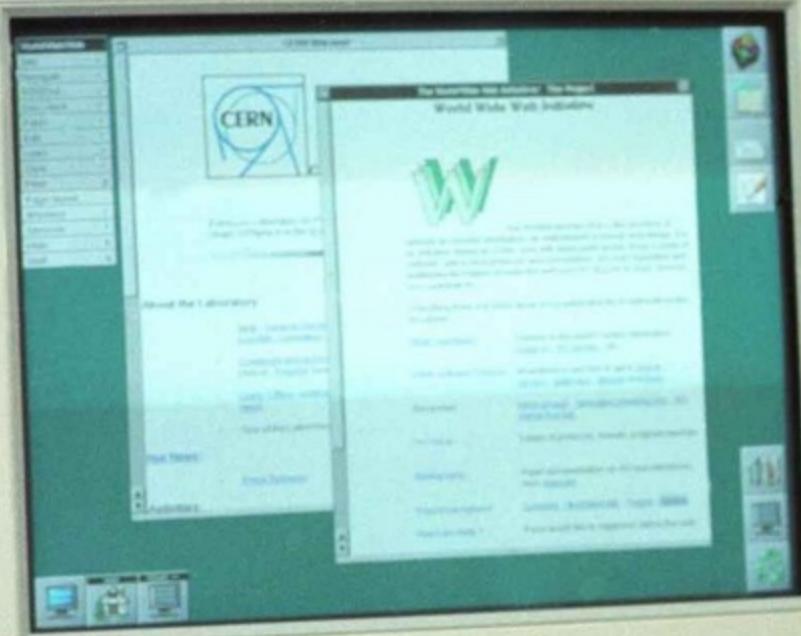


The Large Hadron Collider (LHC)
The largest scientific experiment to date
1 PB / second of raw physics data

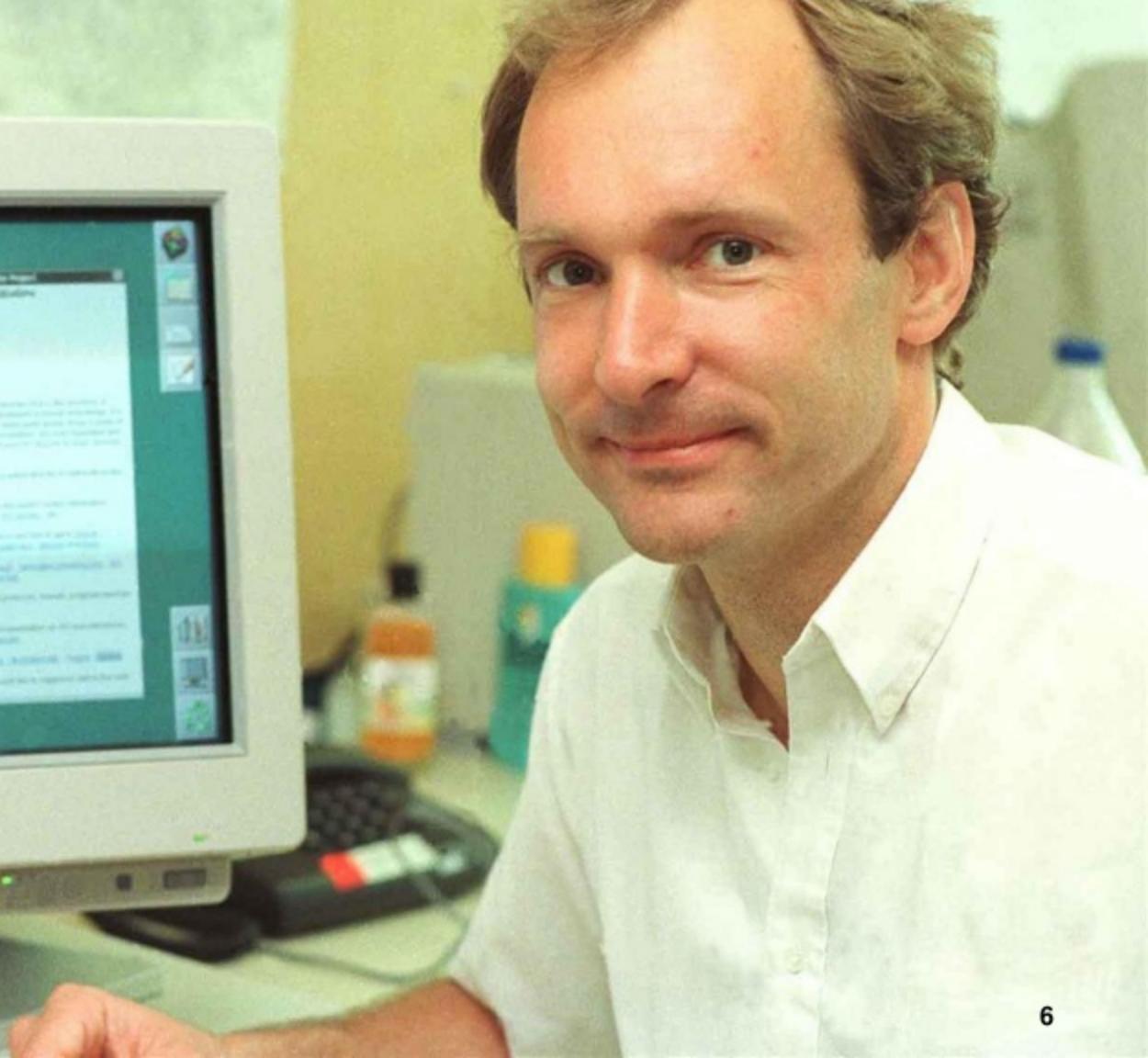




Discovery of the Higgs boson
ATLAS & CMS 2012



Invention of the WWW
Sir Tim Berners-Lee 1989
<http://info.cern.ch>





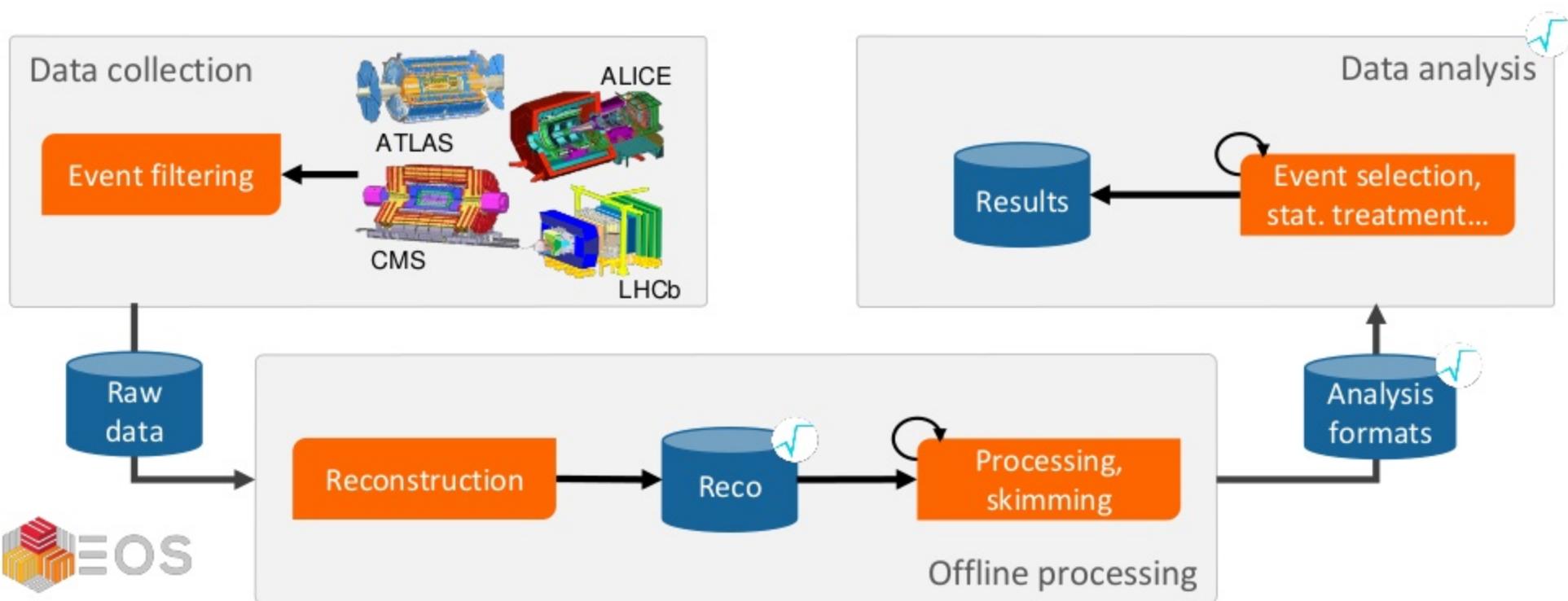
>15,000 scientists
>100 nationalities



Physics Data Processing and Analysis at CERN

#SAISExp1

LHC Data Pipeline at CERN



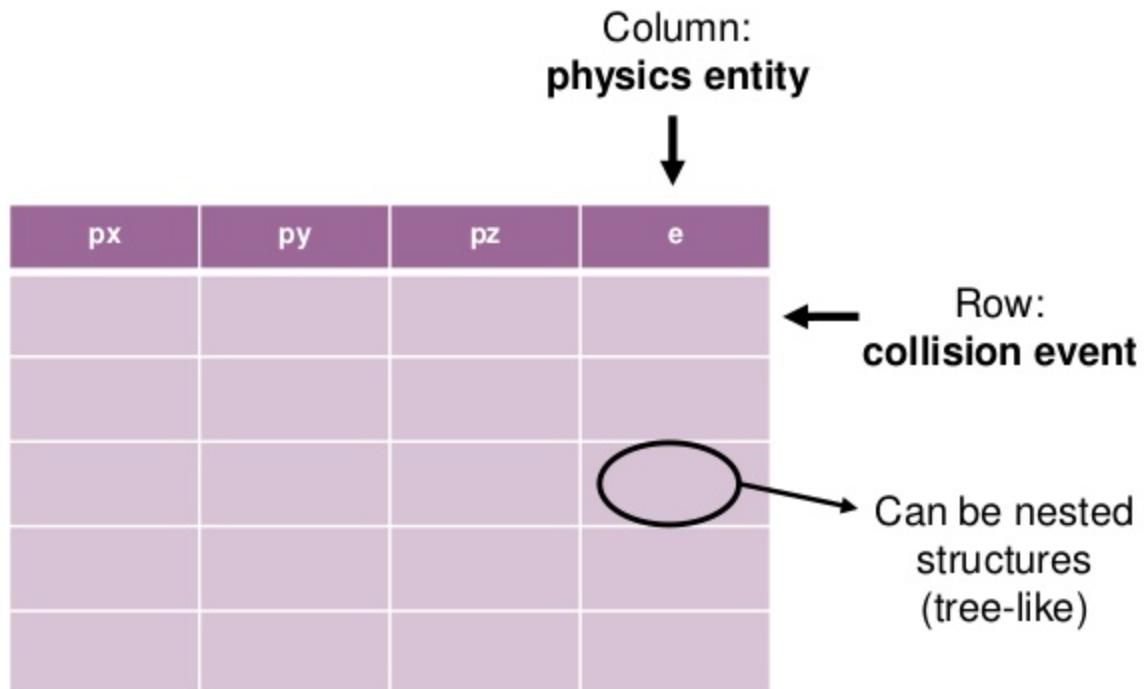
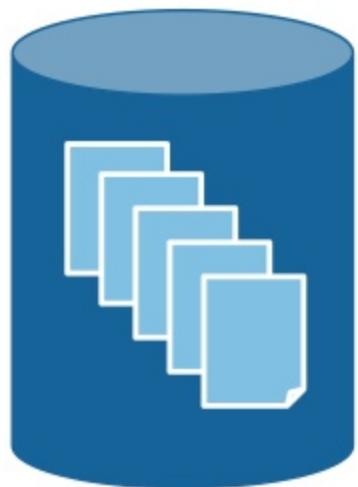
ROOT

- **The** data analysis framework for high-energy physics (HEP)
- Data processing, statistical analysis, visualisation, I/O and storage
- **~1 EB** stored in ROOT format
 - Binary, compressed
 - Columnar



<https://root.cern>

ROOT Dataset



ROOT dataset stored in
one or multiple files

The LHC Computing Grid

Started in 2002

Provide processing power and data access to physicists

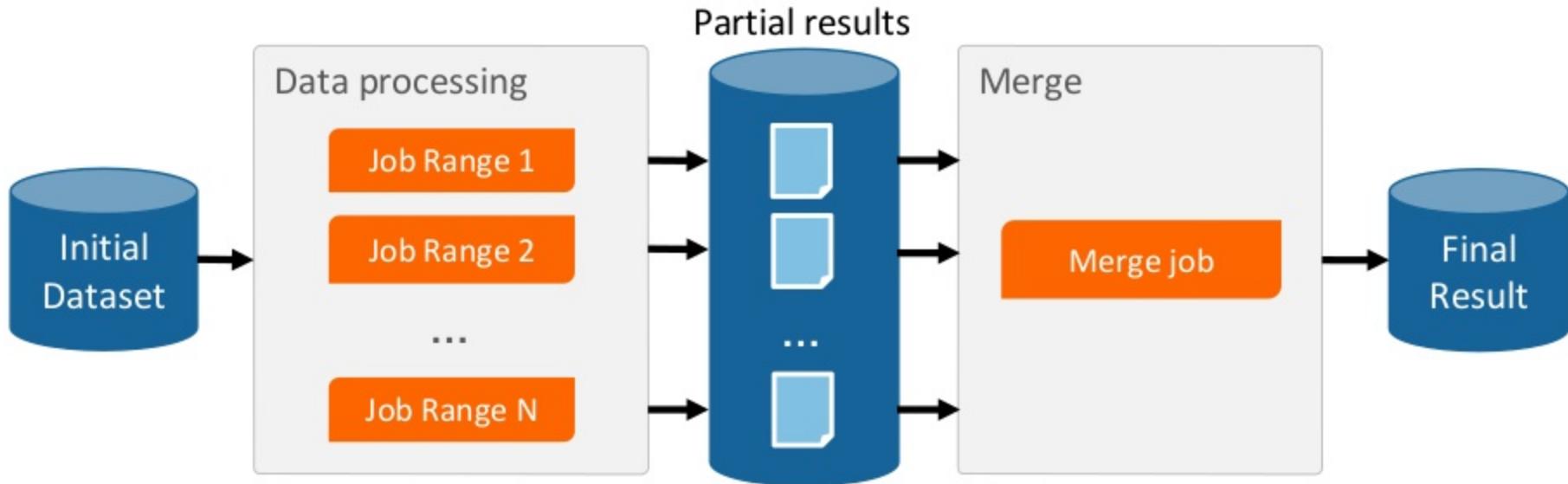
~170 centres in 42 countries

Running 24/7/365



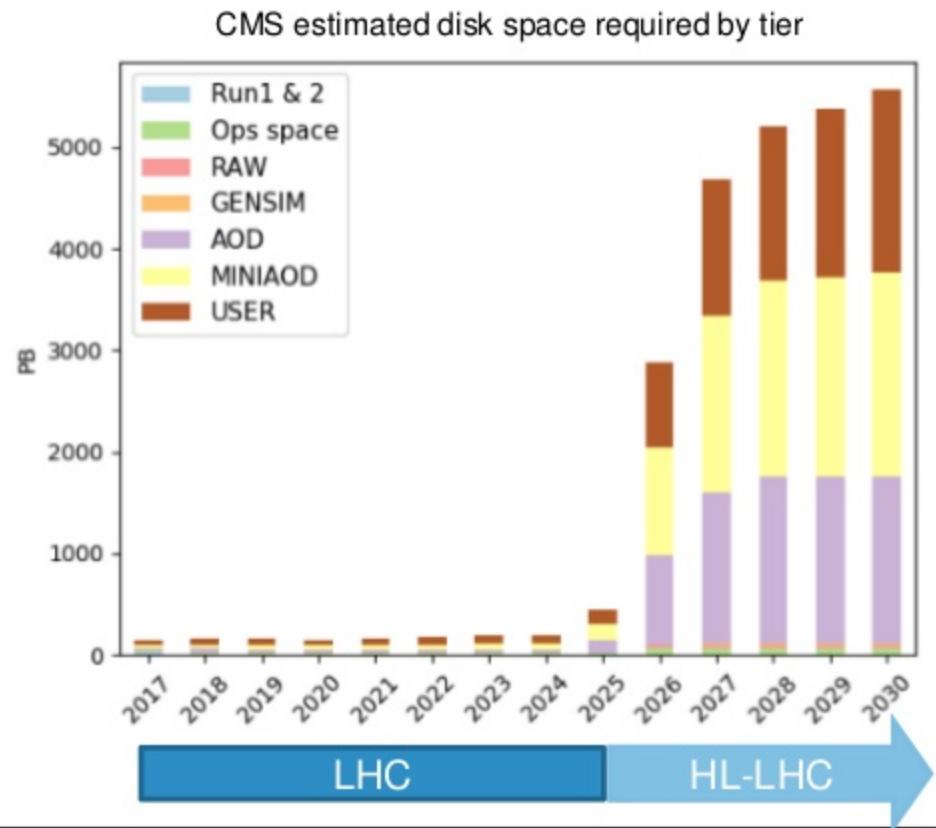
Distributed Computing in HEP

- Physicists use Grid and batch resources to process LHC data in parallel

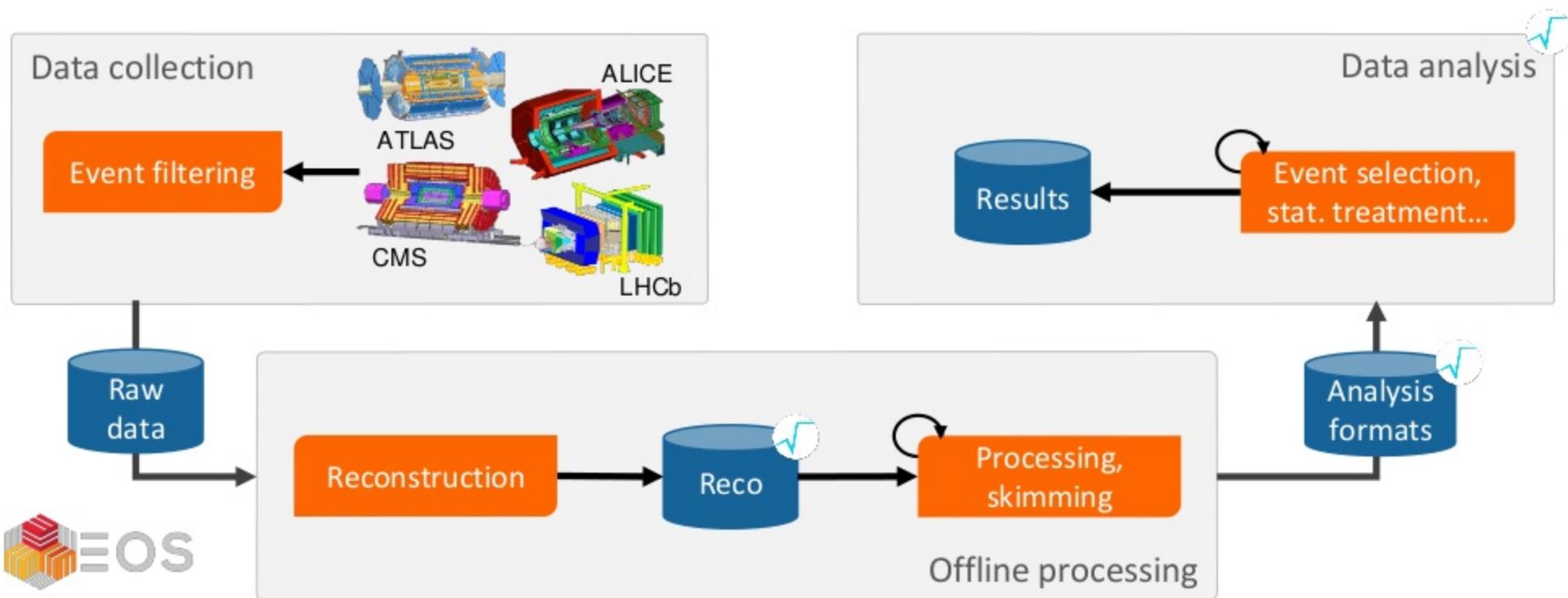


HL-LHC: Even More Data!

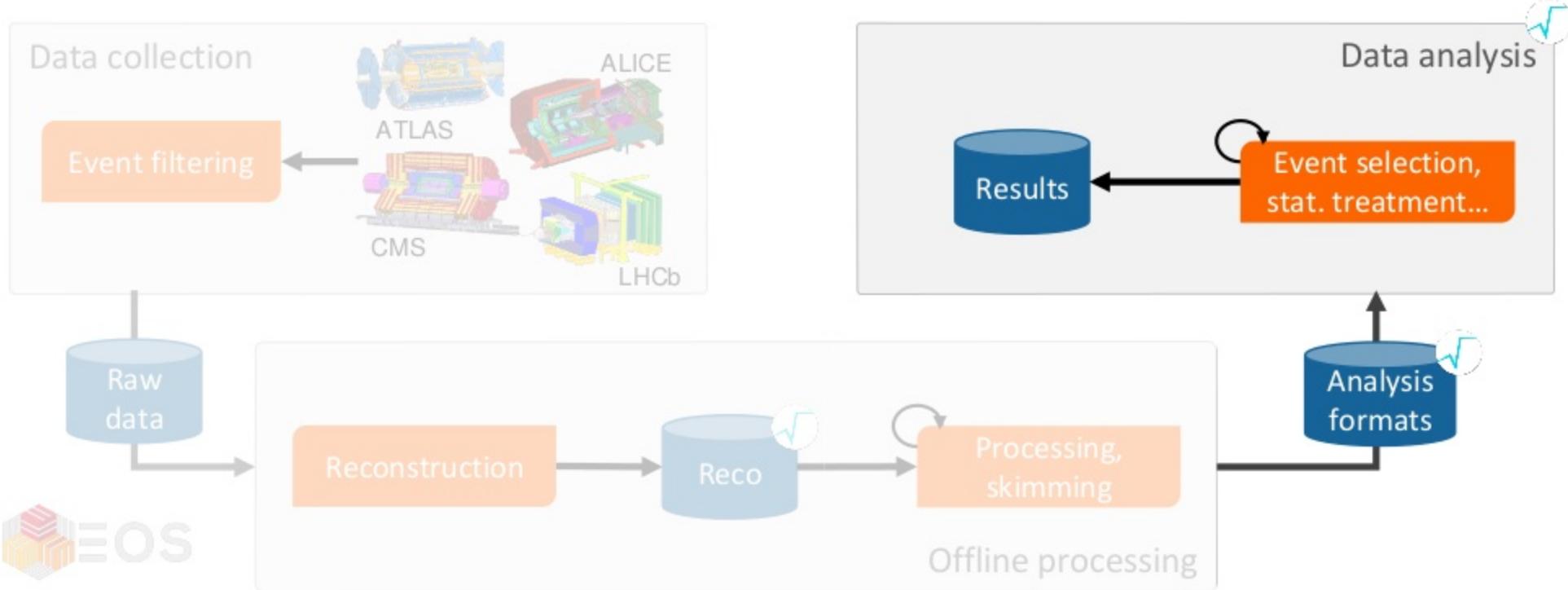
- Coming upgrade:
High-Luminosity LHC
- **30x more data**
collected
- Big challenge for
software and
computing



LHC Data Pipeline at CERN



Final Steps of Data Analysis





Interactive Data Analysis

##SAISExp1

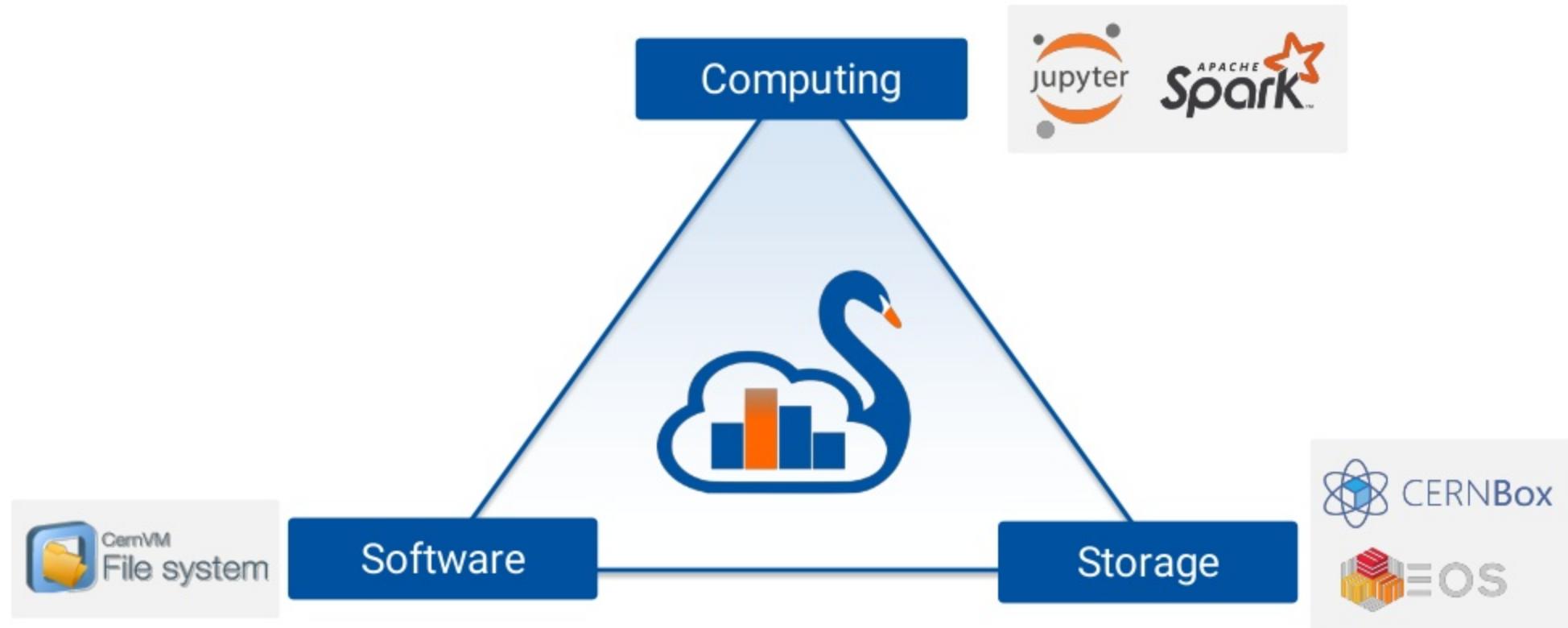
The SWAN Service

- **SWAN:** Service for Web-based Analysis
- **Interactive computing** platform for scientists
 - Based on Jupyter notebooks
- Analysis with only a web browser
- Easy **sharing** of results
- Integrated with CERN resources
 - Storage, software and computing

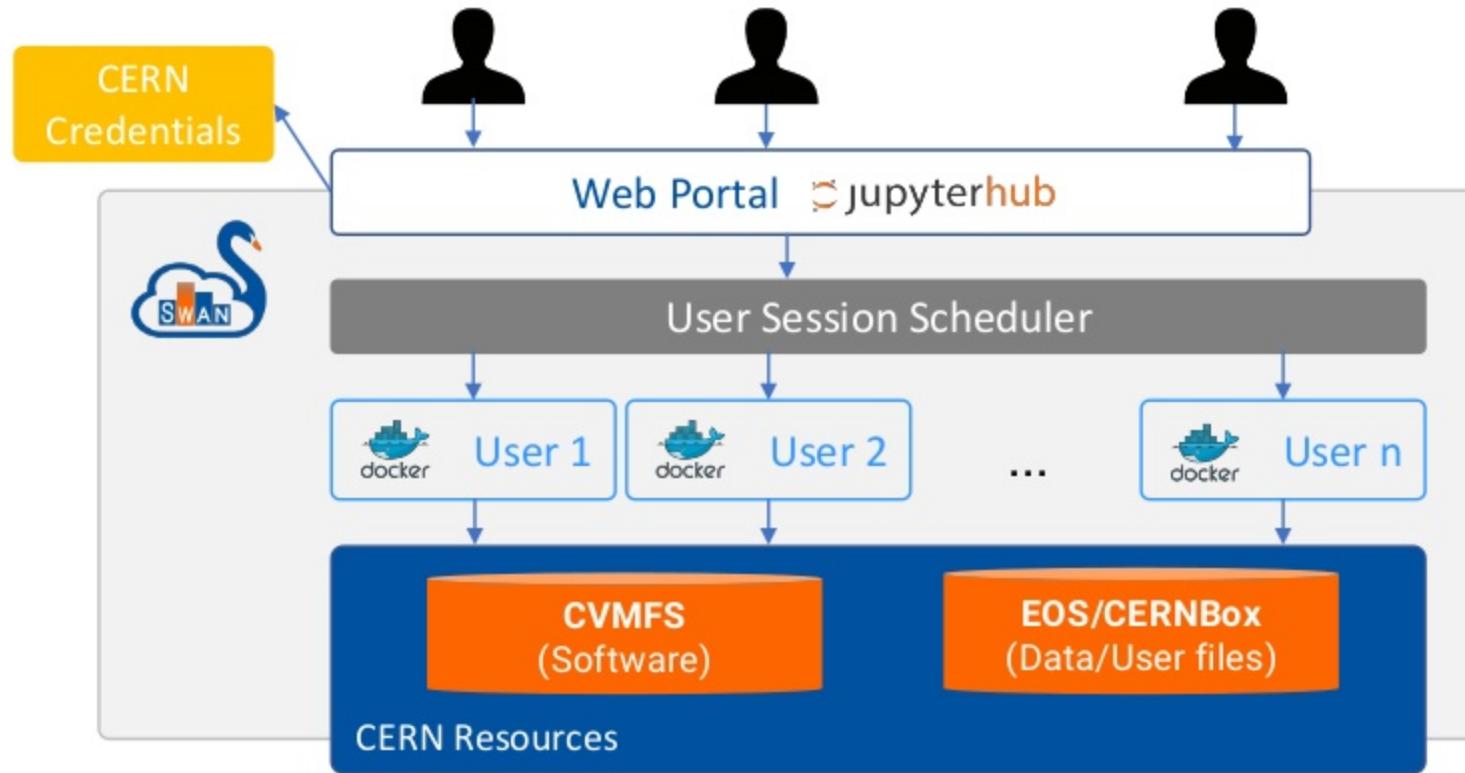


<https://swan.web.cern.ch>

SWAN Pillars



SWAN Architecture



SWAN Interface: Notebooks

SWAN > My Projects

My Projects

NAME	STATUS	MODIFIED
Proj1		5 days ago
Proj2		15 days ago
Project		21 days ago
Project 1		2 months ago
Project 2		4 months ago
ProjTest		15 days ago
Spark		7 days ago
SWAN-Spark_NXCALS_Example		20 days ago
teste		19 days ago

SWAN © Copyright CERN 2017. All rights reserved.
[Home](#) | [Contacts](#) | [Support](#) | [Report a bug](#) | [Imprint](#)



Simple_ROOTbook.cpp.pymb

Simple ROOTbook (C++)

This simple ROOTbook shows how to create a `TH1F` and `draw()`. The language chosen is C++.

In order to activate the interactive visualisation we can use the `!ROOT` magic:

```
In [1]: %pyroot as
```

Now we will create a `TH1F` specifying its title and axes titles:

```
In [2]: TH1F h1("myhisto","My Histowith X axis/Y axis",64, -4, 4)
```

```
[TMRP 1] Name: myhisto Title: My Histowith X axis
```

If you are wondering what this output represents, it is what we call a "printed value". The ROOT Interpreter can indeed be instructed to "print" according to certain rules instances of a particular class.

Time to create a random generator and fill our histogram:

```
In [3]: Random3 rndGenerator
```

```
[for i in range(1000):
```

```
    var rnd = rndGenerator.Gen();
```

```
    h.Fill(rnd);
```

```
]
```

We can now draw the histogram. We will first create a `TCanvas`, the entity which in ROOT holds graphics primitives.

```
In [4]: TCanvas c2
```

```
In [5]: h.Draw()
```



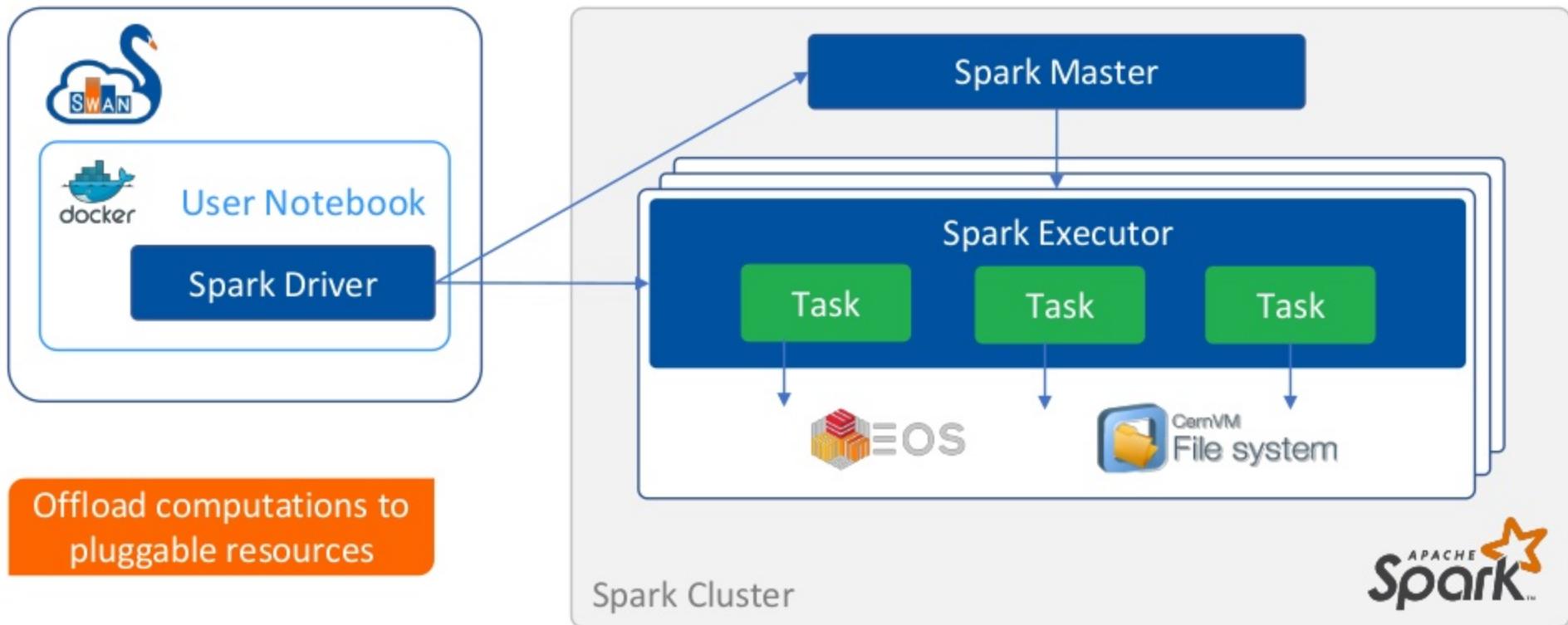
#SAISExp1

Sharing in SWAN

The screenshot shows the SWAN interface with a 'Share Project' dialog open over a project page. The project page title is 'Super Real Analysis with TOTEM data'. The dialog title is 'Share Project' and it displays the message 'You are sharing: Super Real Analysis with TOTEM data'. It includes a search bar with placeholder 'Start typing to add names...' and a list of users shared with: 'Danilo Piparo (danilo)' and 'Enric Tejedor Saavedra (enric)'. At the bottom are 'Stop Sharing' and 'Update' buttons.

The screenshot shows the SWAN interface with the 'Share' tab selected. The top navigation bar includes 'Projects', 'Share', and 'CERNBox'. The main content area has two sections: 'Projects shared with me' and 'Projects shared by me'. The 'Projects shared with me' section lists one item: 'UP2University Pilot' (Shared by jupytercon2, 7 minutes ago). The 'Projects shared by me' section lists two items: 'Higgs Boson discovery' (Shared with 2 people/groups, 18 hours ago) and 'Super Real Analysis with TOTEM data' (Shared with dogo, 19 hours ago). Both sections include columns for NAME, SIZE, SHARED BY, and DATE.

Integration with Spark



Spark Connector

The screenshot shows a Jupyter Notebook interface with the title "Spark > physics_analysis_using_swam_spark_template". The notebook header includes a cloud icon, the title, and a "Not Trusted" status. Below the header is a toolbar with various icons. The main content area features the SWAN logo (a blue swan on a cloud) and the Apache Spark logo. A section titled "Integration of SWAN with Spark clusters" contains text about the functionality provided by a SWAN prototype machine. It includes a code cell (In [1]) with Python code for generating credentials:

```
import getpass
import os, re

print("Please enter your password")
ret = os.system("echo '%s' | xterm -e re.escape(getpass.getpass())")

if ret == 0: print("Credentials created successfully")
else:      sys.stderr.write('Error creating credentials, return code: %s\n' % ret)
```

The screenshot shows a Jupyter Notebook interface with the title "Spark > Spark_Simple". The notebook header includes a cloud icon, the title, and a "Not Trusted" status. Below the header is a toolbar with various icons. The main content area features the Apache Spark logo and a section titled "Simple example with Spark". It contains text about the use of Spark in SWAN and its capabilities. An orange callout box in the bottom right corner says "Configure Spark and connect to cluster with a click". To the right of the notebook is a configuration panel titled "Spark clusters connection" with the sub-section "Selected configuration". It lists configuration options like spark.shuffle.service.enabled, spark.driver.memory, and spark.executor.instances. A green "Connect" button is at the bottom of the configuration panel.

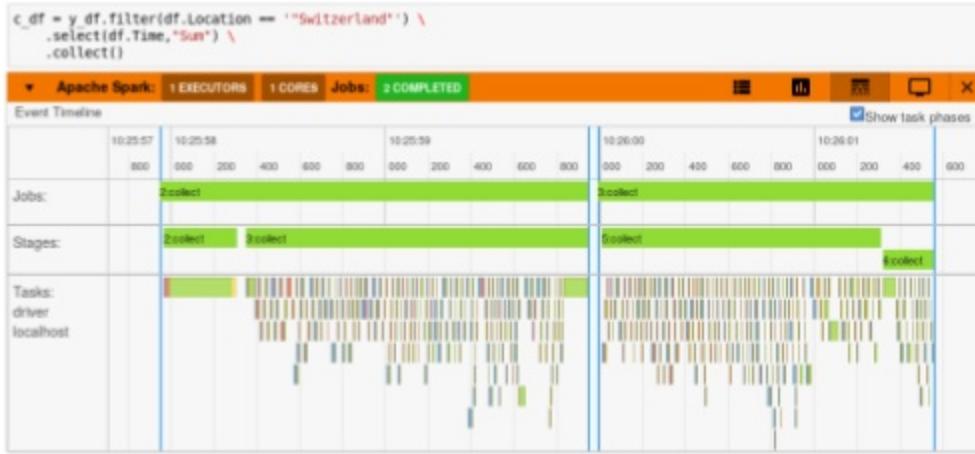
Spark Monitor



[Code here!](#)

Google Summer of Code

- Bridge the gap between interactive computing and distributed data processing
- Automatically appears when a Spark job is submitted from a cell
- Progress bars, task timeline, resource utilisation



Job ID	Job Name	Status	Stages	Tasks	Submission Time	Duration
2	reduce	COMPLETED	2/2	48 / 48	5 minutes ago	3s
5	reduce	COMPLETED	1/1	30 / 30	5 minutes ago	2s
4	coalesce	COMPLETED	1/1	30 / 30	5 minutes ago	0s
3	foreach	COMPLETED	1/1 (1 skipped)	30 / 30	5 minutes ago	1m 20s
6	coalesce	PENDING			Unknown	-
7	foreach	COMPLETED	1/1	30 / 30	5 minutes ago	1m 20s



Physics Data Use Case

#SAISExp1

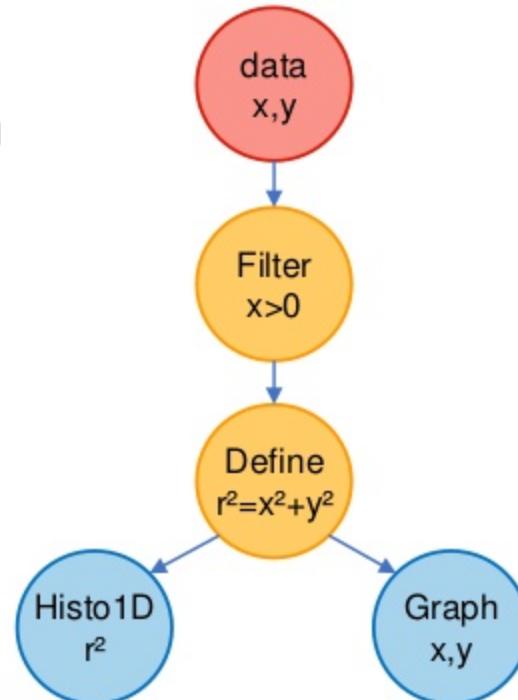
The HEP DataFrame

- **RDataFrame**
 - Implemented in C++, interfaced also to Python
 - Tailored for ROOT and HEP

```
df = RDataFrame(dataset)

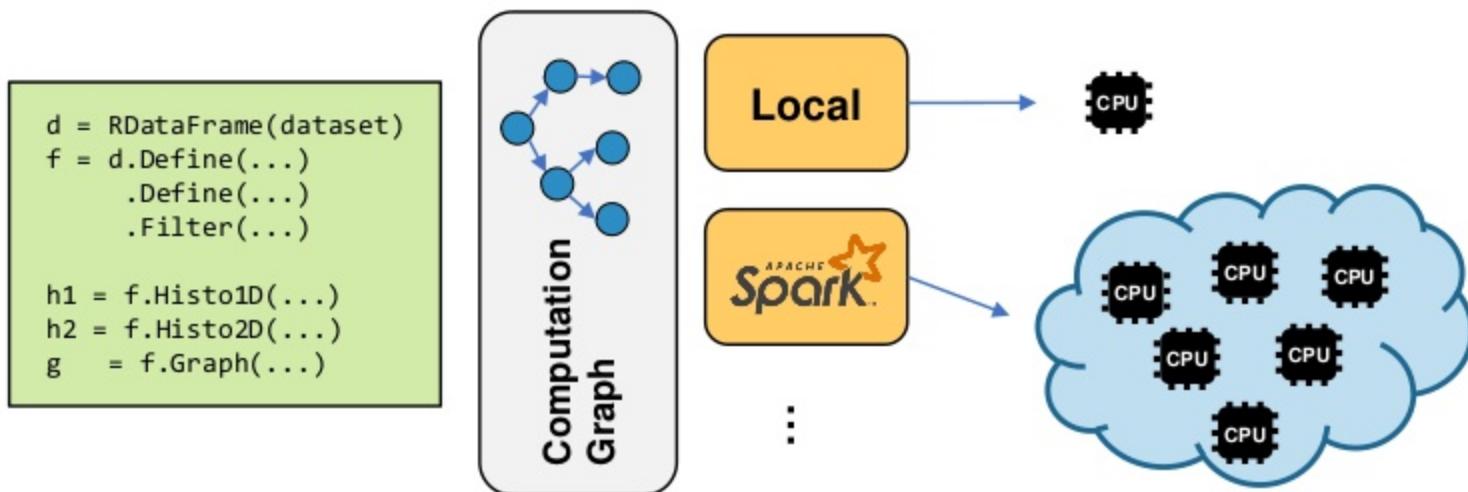
df2 = df.Filter('x > 0')
      .Define('r2', 'x*x + y*y')

h = df2.Histo1D('r2')
g = df2.Graph('x', 'y')
```



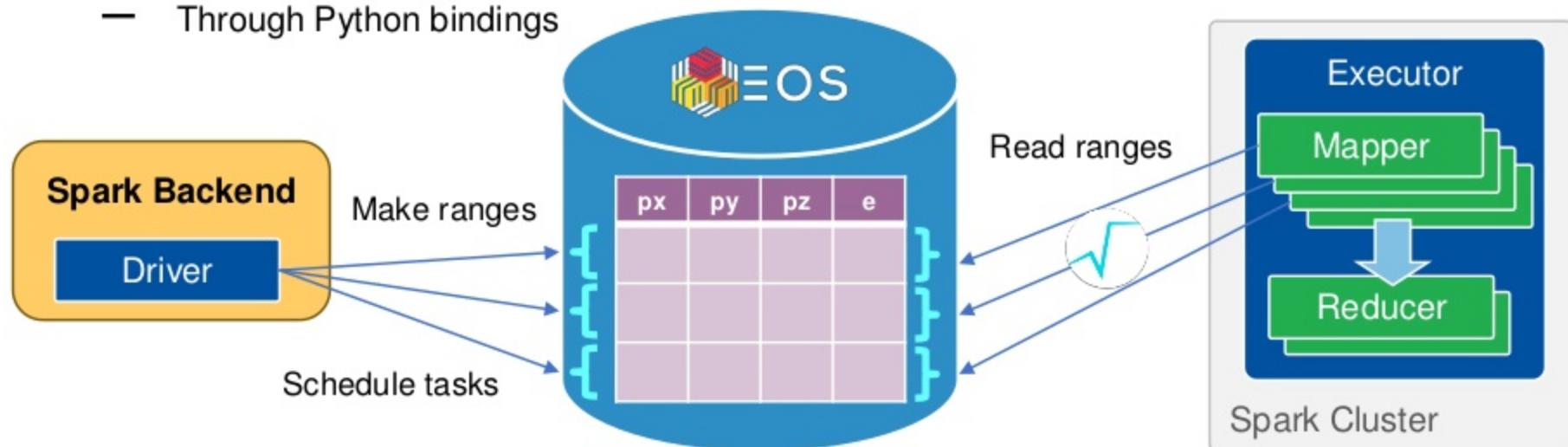
Distributed RDataFrame

- Exploratory work to parallelise RDataFrame computations with multiple backends



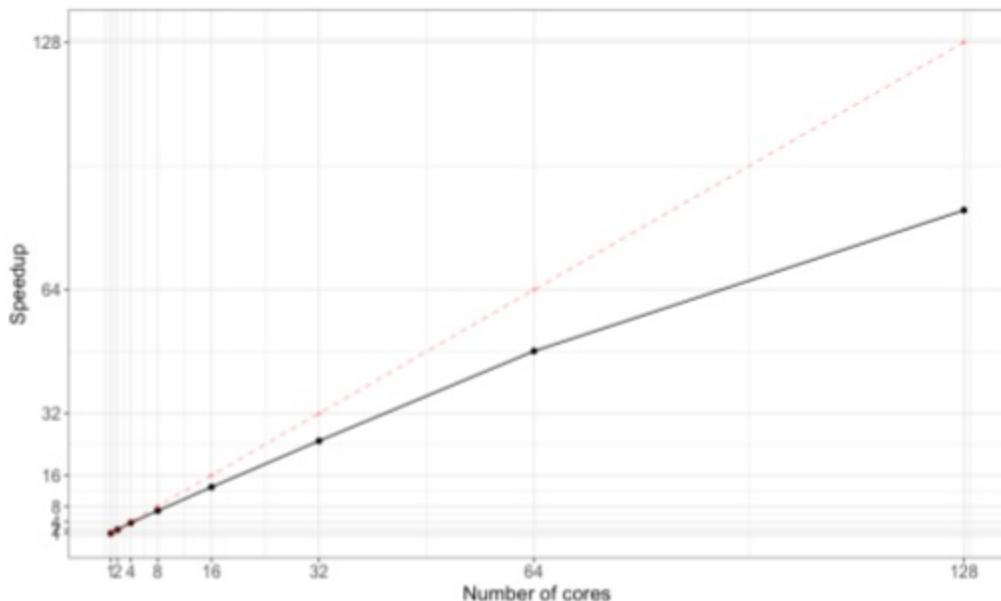
Spark Backend for RDataFrame

- **Map-reduce** workflow where every mapper runs the RDataFrame computation graph on a **range** of collision events
- Run analysis in **C++** with Spark
 - Through Python bindings



Real Example: TOTEM Analysis

- TOTEM experiment analysis coded with RDataFrame
- **Spark** backend
- **4.7 TB** dataset on EOS
- Launched from **SWAN** to a dedicated Spark cluster
- **Get to physics results faster!**





Controls Data Use Case

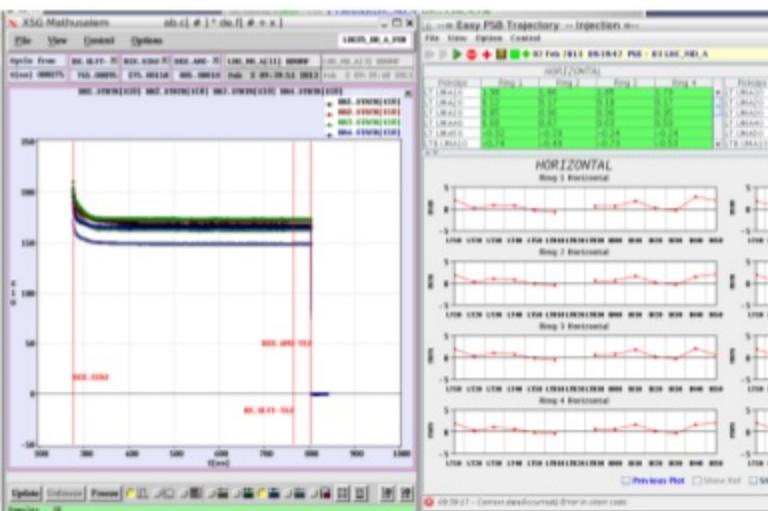
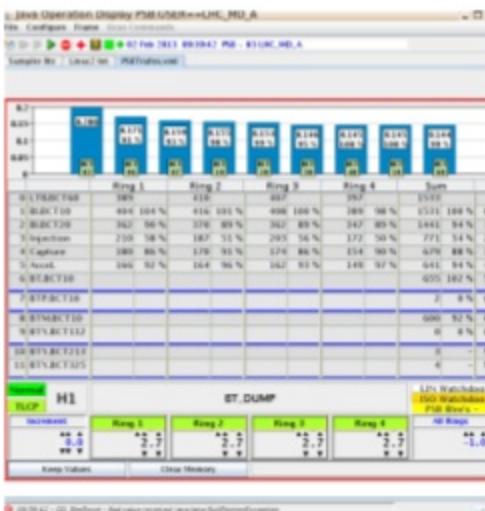
#SAISExp1



LHC: Huge machine, highly sophisticated
Control and monitoring are crucial

Hardware Controls

- Complex control system for monitoring
- 1000s of devices, 100s of properties each



This screenshot shows the Easy PSB Trajectory software interface. It has a header bar with menu options like File, Edit, View, References, Commands, Control, Programs, and Help. Below the header is a toolbar with icons for file operations. The main area contains two tables. The first table, titled 'Simple view', lists components with columns for 'POW', 'Status', 'CCV', 'ACN', and 'Unit'. The second table, titled 'POW', also lists components with similar columns. At the bottom, there's a table for 'PTIM-V' with columns for 'Future', 'CCV', 'ACN', 'Start', and 'Train'.

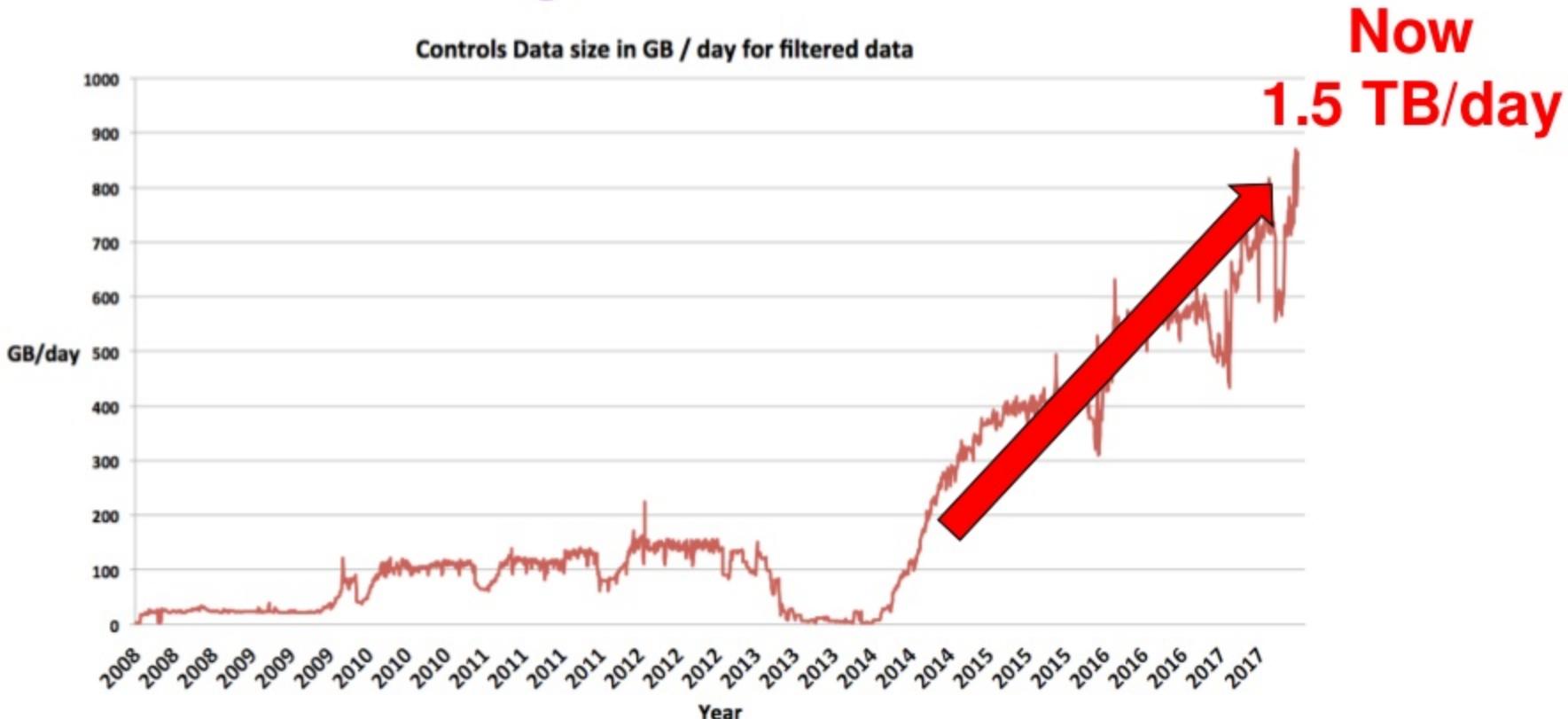
POW	Status	CCV	ACN	Unit
SPB1_0N0412L3	0e	1.07	0.14	A
SPB2_0N0412L3	0e	-0.84	-0.47	A
SPB3_0N0412L3	0e	-1.09	-0.94	A
SPB4_0N0412L3	0e	1.31	0.99	A
SPB1_0N0016L3	0e	-2.63	-1.85	A
SPB2_0N0016L3	0e	-3.69	-3.26	A
SPB3_0N0016L3	0e	-2.69	-1.66	A
SPB4_0N0016L3	0e	-1.77	-0.86	A

POW	Status	CCV	ACN	Unit
SPB1_0N0413	0e	0.11	0.15	A
SPB2_0N0413	0e	0.91	0.05	A
SPB3_0N0413	0e	7.40	7.43	A
SPB4_0N0413	0e	7.04	6.20	A
SPB1_0N0013	0e	1.06	0.77	A
SPB2_0N0013	0e	6.83	5.97	A
SPB3_0N0013	0e	6.83	20.80	A

POW	Status	CCV	ACN	Unit
SPB1_XSH2L4	0e	-5.28	-4.89	A
SPB2_XSH2L4	0e	0.51	0.31	A
SPB3_XSH2L4	0e	-3.07	-2.66	A
SPB4_XSH2L4	0e	2.99	2.66	A
SPB1_XSH4L2	0e	-1.19	-1.92	A
SPB2_XSH4L2	0e	-0.26	-0.15	A
SPB4_XSH4L2	0e	16.40	16.75	A
SPB1_XSH6L2	0e	12.77	12.25	A
SPB2_XSH6L2	0e	9.06	8.18	A
SPB4_XSH6L2	0e	17.00	10.10	A
SPB1_XSH8L4	0e	7.12	6.81	A
SPB2_XSH8L4	0e	1.93	1.68	A
SPB3_XSH8L4	0e	2.92	2.85	A
SPB4_XSH8L4	0e	0.39	1.25	A

PTIM-V	Future	CCV	ACN	Start	Train
SPB1_ADXAP	Enabled	10	10	80% HSD-CT	2-190
SPB1_ADXET	Enabled	25	25	80% HSD-CT	2-190
SPB1_ADXCT	Enabled	10	10	80% HSD-CT	2-190
SPB3_ADXK	Enabled	356	356	80% SCY-CT	2-190

Controls Log Data Growth



CERN Accelerator Logging Service

- Old system based on SQL databases
 - Hard to scale horizontally
 - Slow data extraction
- New system (NXCALS)
 - Data pumped into HBase and HDFS (Parquet)
 - **Spark** to extract and process data
 - **SWAN** to visualise + analyse



NXCALS Data Analysis in SWAN

- NXCALS rely on SWAN as their data analysis platform
- Connection to Spark clusters
- Access to software (data science Python ecosystem)

Inspect data

```
In [2]: df1.select('acqStamp','voltage_18V','current_18V','device','pt100Value').toPandas()[:5]
```

```
Out[2]:
```

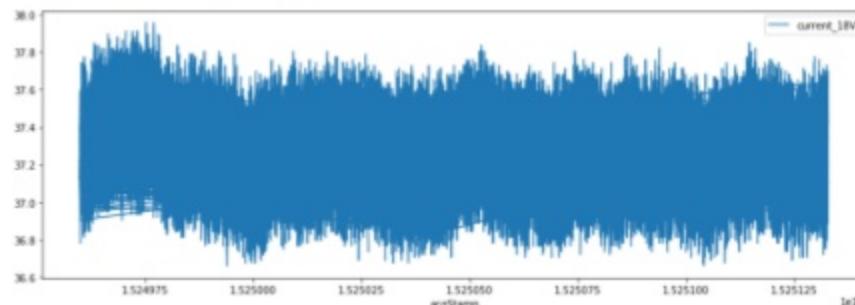
	acqStamp	voltage_18V	current_18V	device	pt100Value
0	15249601031328650000	NaN	37.301794	RADMON.PS-10	106.578911
1	15249602841342540000	NaN	37.246742	RADMON.PS-10	107.246742
2	15249603221349420000	37.560940	37.560940	RADMON.PS-10	106.504707
3	15249603331352440000	20.099966	NaN	RADMON.PS-10	107.069654
4	15249609111425480000	20.111261	37.698135	RADMON.PS-10	106.578911

Draw a plot with matplotlib

```
In [3]: import matplotlib
import pandas as pd
%matplotlib inline
```

```
In [4]: p_df = df1.select('acqStamp','current_18V').toPandas()
p_df.plot('acqStamp','current_18V',figsize=(15,5))
# p_df.sort_values(by='acqStamp').plot(pd.to_datetime(p_df['acqStamp'].unit='ns'),'current_18V',figsize=(15,5))
```

```
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd8fa2bcc>
```





Summary

#SAISExp1

Challenges

- The increase in physics and controls data volumes and complexity is **pushing software** at CERN
 - Adoption of Spark and other big data technologies still in its early stages
- **Large codebase** developed over decades
 - Cannot change overnight
- Changing the **mindset** of programmers takes time
 - Declarative analysis
 - Pushing computations to data

Future Directions

- **Bridge the gap** between data processing needs and technology evolution
 - Complement traditional ways with new strategies
- Combine **interactive analysis** with **easy access** to more processing power
 - Higher-level programming models
 - Pluggable computing resources

More on CERN and Spark: stay tuned for [Luca's](#) and [Prasanth's](#) presentations



Backup Slides

#SAISExp1

JupyterLab

- Jupyter is evolving towards a desktop-like environment
 - Notebook, terminal, file browser, editors, ...
 - Highly customisable

