SWAN for NXCALS

Piotr Sowiński, BE-CO-DS, 11.10.2019
on behalf of NXCALS Team

Agenda

- Accelerator Logging Service
- NXCALS Project
- SWAN for NXCALS
- User feedback
- Conclusions

Background:
CERN Accelerator Logging Service

- CALS is a **CERN-wide mission-critical service** in production since 2003.
- Stores >2TB/day of data from >2.6 million signals of accelerator equipment and beam measurements. ~1PB of data logged so far.
- Used by **>1000 users** from across CERN.
- Currently based on Oracle DBMS.
- High Level Objectives:
  - Information management for **accelerator performance improvement**
  - Make available **long term statistics** for management and help **decision support** process
  - **Avoid duplicate logging** efforts

NXCALS Project

Motivation:

- In general, accelerator operation has stabilised
- Increasing number of complex data analysis use cases, looking at longterm data sets with a goal of tuning performance and preparing for the future (HL-LHC and beyond).
- CALS system has reached its limits when it comes to data analysis functionality.

- NXCALS is the **successor** of CALS based on Hadoop Big Data technologies using cluster computing power for data analysis.
- The horizontally-scalable architecture should be capable of confronting **highly increasing amount of logged data.**

NXCALS Team

Jakub Wozniak
(IC staff)

Piotr Sowinski
(IC staff)

Kamil Krynicki
(Fellow)

Marcin Sobieszek
(LD staff)

Grigorios Avgitidis
(Fellow)

Viktor Hryhyniak
(TS)

In Collaboration With…

IT-DB & EP-SFT

- Hadoop BigData storage software & hardware platform
- SWAN as a Data Science platform with NXCALS

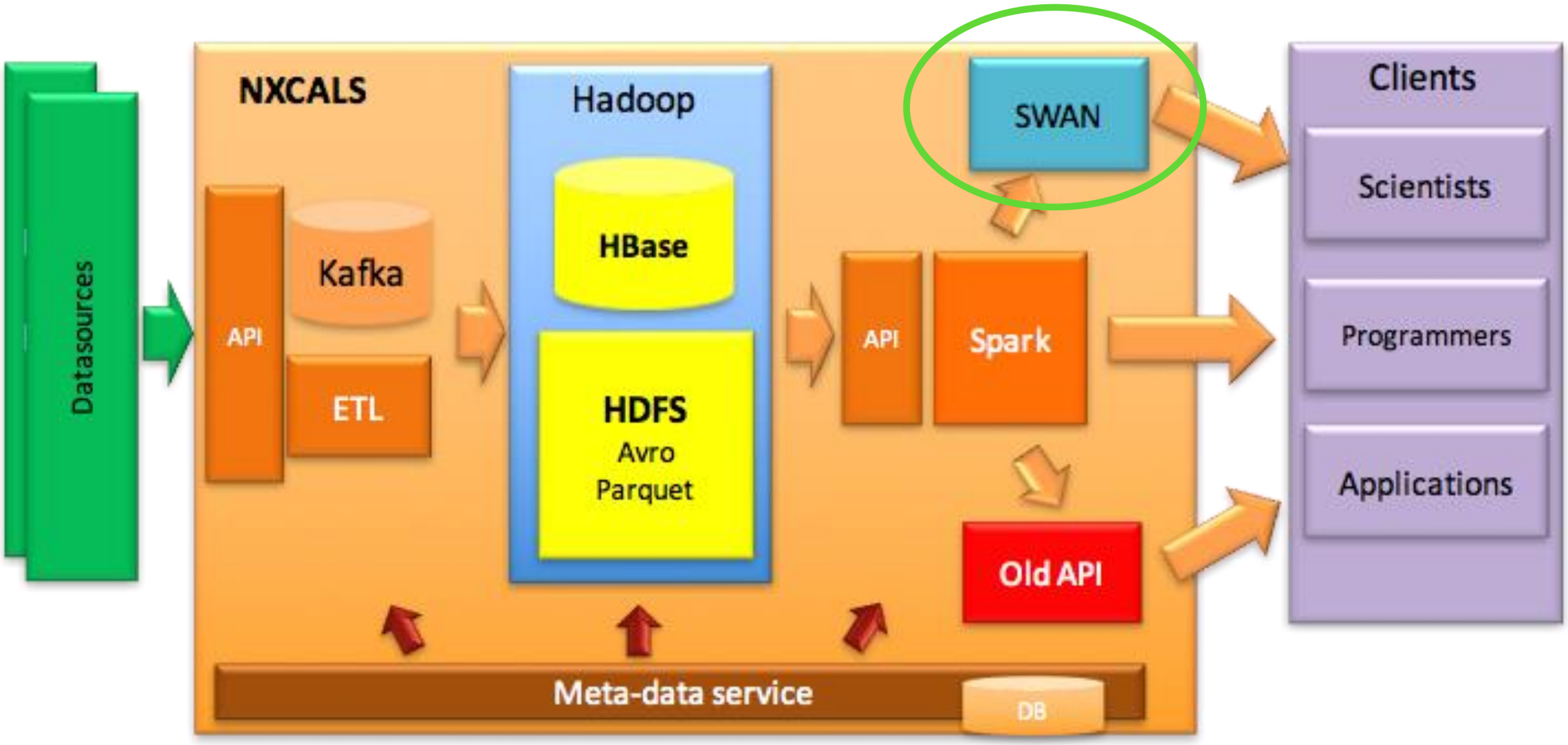
TE-MPE

- To have a single data archive for Post-Mortem using NXCALS

BE-ICS

- On WinccOA to NXCALS datasources

NXCALS Architecture



How SWAN Fits Whole Picture ?

- Proposed as one of the **NXCALS data access methods** among Java API, NXCALS Spark bundle and others

- Thanks to joint effort between BE-CO, IT and EP-SFT, **NXCALS is well integrated into SWAN environment:**

  - New **NXCALS releases are automatically deployed** - on a weekly basis NXCALS libraries are collected from the dedicated EOS space.

  - SWAN team ensures the software stack for NXCALS contains correct Python and Spark versions + required libraries

- Python **Data Access API easily accessible** offering different data query builders

- Plays **important role especially during the CALS -> NXCALS transition** period (testing, ad-hoc queries, TIMBER substitute). Being **used on a daily basis**.

- More details can be found in **NXCALS Documentation**: http://nxcals-docs.web.cern.ch

NXCALS Data After Few Clicks !

- After requesting authorisation to NXCALS service it is sufficient to:
  - Provide CERN credentials
  - Select Environment (NXCALS Python3 software stack and BE NXCALS Spark cluster)
  - Establish Spark clusters connection (with bundled NXCALS configuration)
  - Import NXCALS builders and execute a code as in the example below:

```
In [3]: from cern.nxcals.pyquery.builders import *

df1 = DevicePropertyQuery.builder(spark)\
    .system('CMW').startTime('2017-08-29 00:00:00.000').duration(10000000000)\
    .entity().parameter('RADMON.PS-10/ExpertMonitoringAcquisition')\
    .buildDataset()
```

Apache Spark: 2 EXECUTORS 4 CORES Jobs: 1 COMPLETED

Job ID	Job Name	Status	Stages	Tasks	Submission Time	Duration
1	load	COMPLETED	1/1	1 / 1	2 minutes ago	1s

When to use SWAN ?

- For code prototyping - building SPARK queries, checking performance, etc
- Ad-hoc queries - no additional setup / configuration required
- For the simple visualisation of results using Pandas through matplotlib for example
- Sharing code / analysis results

Users Feedback

- NXCALS user questionnaire (11-03-2019), with a purpose to:
  - Establish contact / start dialog with the NXCALS user community using SWAN
  - Better understand NXCALS user needs concerning the service
  - Suggest improvements and new functionalities to SWAN service as required from the NXCALS perspective
- Areas covered include: **Usage, Expectations, Satisfaction**

Survey Responses - Usage

- How often is it used ?
  - From “Occasional” to “On the regular basis”
  - Daily / Weekly
- How is it actually used ?
  - Sharing scripts or studies
  - Hosting documentation in the notebooks
  - Testing NXCALS *(quick and easy, before proper coding)*
  - Plotting and charting *(no NXCALS TIMBER (GUI) yet, plenty of Python libraries)*
  - Ad-hoc queries / scrapbook (post processing, machine development on-line analysis, prototyping code)

Survey Responses - Expectations

- How critical the service availability is for your activities ?
  - Mixed answers from “Critical” to “Non Critical” due to software instabilities *(b.t.w which is much improved since then)*
- Does it have to be up 24/7 ?
  - It is almost the case already now
  - Expected to be up during working hours, but often outside working hours
- What should be reaction time in case of issues ?
  - Replies range from 1 hour to 1-2 days…

Survey Responses - Satisfaction

- How easy is the service to use currently ?
  - Very easy to use (strongest point of the service)
- How fast is the SWAN service in providing you the required results ?
  - Fast enough / Not as fast as a desktop, but totally acceptable so far for my activities in general *(processing on the client side might be faster but running Spark application in Hadoop is the same)*
- Are you satisfied with integration of NXCALS services ?
  - The limitation to only a single Spark connection per session is a severe limitation *(this limitation has since been removed)*
  - The connection time to Hadoop/NXCALS is very slow *(currently much faster)*

Conclusions

- Well integrated with NXCALS, being an easy “access point” to logging data
- Always up to date thanks to automatic NXCALS libraries deployment
- As reported by many users - very useful for data analysis, plotting and sharing which helps during transition period from CALS to NXCALS when new TIMBER GUI application will be developed
- Excellent tool for code prototyping
- Very good cross-department collaboration, resulting in end-user issues being efficiently resolved, in particular:
  - Reliability has improved a lot.
  - Functionality to have multiple session was introduced.

Thank you !

Main Team Focus In 2019 and Onwards

- NXCALS contains all of the historical CALS data
  - Migration is progressing well
    - Dedicated setup deployed on 65 machines
    - 1PB to be moved, including tens of corner cases from CALS and CCDB (but ~95% of data already moved. That represents 25 x 10^12 of data points !)
- A backward compatible Java client API is provided
  - Old CALS API has 126 methods
  - Progressive releases being prepared for early adopters
- A new version of Timber is available in production
- The data extraction performance is at least comparable with CALS

Decommission CALS by the end of LS2 once the above criteria have been satisfied