SWAN integrating powers
storage, compute, interactivity, collaboration

Jakub T. Mościcki
On behalf of the SWAN team

https://cern.ch/swan

Nov 4th, 2019
CHEP Adeleide
Yet Another Jupyter Service?

How SWAN is different and does it fly?
Web-based Analysis

Ready-to-go environment
“one click away”

Software

CVMFS
LHC experiments
Machine Learning
100s libraries
...

CERNBox
Personal & project files
Sharing&synchronization

EOS
Big data repositories
Online access

Storage

Compute

Scalable, private computing cloud.

Rich, reproducible environment.

S3/CVMFS, CHEP Track 8,
E.Bocchi, Thu, 2pm, R8

Multiple Jupyter Kernels
Interactive containers
GPU resources
Apache Spark
Condor jobs
Managed K8s clusters

CERN Storage, CHEP Track 8, L.Mascetti Thu, 12.15pm
Collaboration

≠ ivory tower of technology
== using state-of-the-art technology

Users

Designed with users and for users
• New features (libs, kernels, ...)
• Usability improvements...

+2000 users at CERN
4 LHC experiments
20 smaller experiments

Experimental Physics & IT Departments at CERN

EP-SFT Software for Experiments
IT Storage
IT Databases

Cross-institutional

ScienceBox, CHEP Track 7, E.Bocchi Tuesday, 5pm
CS3APiS, CHEP Track 7, H.Labrador, Monday, 2.30pm
SWAN users at CERN

Courses / Training events

2,000+ unique users at CERN in last 6 months
SWAN users at CERN

Users by Department

Experimental Physics

Beams

IT

Director General

Users by Experiment

ATLAS

CMS

LHCb

ALICE

CERN
How is SWAN used?

- SWAN usage at COMPASS
- SWAN for HGCAL Beam Tests Analysis
- My SWAN experience as an ALICE analyzer
- ALICE Data Analysis without wired PC feat. SWAN
- SWAN as a tool in Atlas TDAQ operations
- Integrating CMSSW in SWAN
- AWAKE Data Analysis on SWAN

Summaries from 1st SWAN Workshop, October 2019
https://indico.cern.ch/event/834069
How is SWAN used?

- **SWAN for machine studies**
  - LHC & beams
  - Application of SWAN to the LHC Signal Monitoring Project

- **SWAN for NXCALS -- CERN Accelerator Logging Service**
  - TE/ABT experience with SWAN

- **General Engineering**
  - Automatic generation of superconducting magnet input files for the LEDET software
  - Using SWAN to Analyse Tape Server Logs

- **Operational radiation protection and possible improvements**

- **Statistical Methods** for the LHC
How is SWAN used?

- Usage of SWAN with the CERN Open Data portal for education and outreach
- Research Environments
- Education & Outreach
- Towards EduSWAN

Open Data Science Mesh, CHEP Track 8, Tue 2.30pm

CERNBox & Malt, CHEP Track 8, H.Labrador, Monday 12.00
What the CERN users say?
Use cases for SWAN

Two types of debugging:

(1) live! There’s an issue NOW. Probably just need to load in a small amount of recent “bad” data, maybe some previous “good” data and investigate as many relevant variables as possible

Here, the interactive visualization is the best part – can see ALL the possibilities in one place, with the exact code that made them – discuss with others, etc.

https://atlas.cern/updates/atlas-news/symphony-atlas

(2) post-run: Something looked a little off last run. Did it look off the run before? When did it start? What else changed? Could need to load a large amount of data, overlay many runs…

Only need to load data once – and can then play around with many plots, without having to wait for the data to load again

And then another use, for post-mortem of new changes: want to document effects of updates on the system and verify that changes had the expected outcome

e.g. new trigger had the expected rate and the CPU changed accordingly

---

SWAN as a tool in TDAQ

Heather Russell, McGill University

11 October 2019

SWAN as a tool in Atlas TDAQ operations – Heather Russel
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:

```python
from cern.nxcs.pyquery.builders import *
```

```
In [3]:
df1 = DevicePropertyQuery.builder(spark)
   .system('CMW').startTime('2017-08-29 00:00:00.000').duration(10000000000)
   .entity().parameter('RADMON.PS-10/ExpertMonitoringAcquisition')
   .buildDataset()
```

11-10-2019

BE-CO-DS

12

SWAN for NXCALS - Piotr Sowiński
How SWAN works for me

- Usage with cernbox and EOS is great
  » Develop locally, run your code both on-line and offline
  » Output is always at hand
Our Experience with SWAN

- Seamless report generation
- No local dependencies
- No installation required
- Integrated access to NXCALS
- Easy to start and collaborate

- Connection/kernel issues
- Limited execution time
- Conversion to a script
- IDE capabilities
- Versioning with GitLab
Possible solution: combined use of SWAN and GitLab

- **Step 1:**
  - Creating a SWAN project

- **Step 2:**
  - Making the created SWAN project locally available (CERNBox client)

- **Step 3:**
  - Creating a GitLab project
  - Setting the folder as local repository for the GitLab project

- **Step 4:**
  - Sharing the GitLab repository

- **Possible improvement:**
  - Integration of Git inside the SWAN service

SWAN for operational radiation protection and possible improvements - Alajos Makovec
Upcoming highlights
NVidia GPU Support - Prototype

> Exploitation of container technologies to provide support for NVidia GPUs
  - Already integrated with ScienceBox (more details soon)

> Prototype server for testing purposes
  - NVidia Tesla V100 PCIe 32GB
  - If interested, ask us to join the beta program

> All the packages are provided by CVMFS
  - Including CUDA enabled machine learning software stack
  - TensorBoard for interactive monitoring

> Collaboration
  - Knowledge Transfer Department
  - IT-CM Group (Laurence Field et al.)
Batch, managed k8s - Prototype

> Batch & Grid jobs submission
  - Monitoring display
  - Jobs tab

> Possibility to connect to user managed Kubernetes clusters
  - Offload Spark computations
  - Control and use your own resources
  - Quickly create, use and dispose
  - Share access with other users
Jupyterlab - Planned

> Next-generation interface for Project Jupyter

- Concurrent editing
- “IDE in the browser”
Contacts
swan-admins@cern.ch
http://cern.ch/swan
https://cern.ch/swan-community

Repository
https://github.com/swan-cern/

ScienceBox
- https://cern.ch/sciencebox
Backup Slides
SWAN in a Nutshell

> CERN’s Jupyter Notebook service

> Analysis only with a web browser
  - No local installation and configuration needed
  - Calculations, input data and results “in the Cloud”
  - Support for: Python (2 and 3), ROOT C++, R and Octave

> What makes it different? The integration with CERN resources
  - Software, storage, mass processing power
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...
  - 91

- **Platform**: more...
  - x86_64-skel-good2-opt

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

- **Number of cores**: more...
  - 2

- **Memory**: more...
  - 8 GB

- **Spark cluster**: more...
  - Hadean/sc2

- Always start with this configuration

Start my Session
2 Displaying graphics

We can now draw the histogram. We will at first create a canvas, the entity which in ROOT hides graphics primitives. Note that thanks to [Jupyter](https://jupyter.org), this is not a static plot but an interactive visualisation. Try to play with it and save it as image when you are satisfied.

```python
In [5]:
c = ROOT.TCanvas()
h.Draw()
c.Draw()
```

We'll try now to beautify the plot a bit, for example filling the histogram with a colour and setting a grid on the canvas.

```python
In [6]:
h.SetFillColor(ROOT.kBlue-10)
c.SetGrid()
h.Draw()
c.Draw()
```
Integrating services
Cloud storage as your Home

- CERNBox is SWAN's home directory
  - Storage for your notebooks and data
  - 16k users and 6PB of user data

- Uses EOS disk storage system
  - All experiment data potentially available
  - 250PB of experimental data at CERN (LHC and others)

- Sync&Share
  - Files synced across devices and the Cloud
  - Collaborative analysis
Sharing made easy

> Sharing from inside SWAN interface
  - Integration with CERNBox
  - List shares from other users

> Users can share “Projects”
  - Special kind of folder that contains notebooks and other files, like input data
  - Self contained
  - Fosters collaboration

> Concurrent editing not supported yet by Jupyter
  - Safer to clone
  - Will be available with Jupyterlab
> Software distributed through CVMFS
  - Distributed read-only filesystem
  - "LCG Releases" - pack a series of compatible packages
  - Reduced Docker Images size
  - Lazy fetching of software
  - Step towards reproducibility (across time and people)

> Possibility to install libraries in user cloud storage
  - Good way to use custom/not mainstream packages
  - Configurable environment
Integration with Spark - Production

> Connection to CERN Spark Clusters
  - Spark: general purpose distributed computing framework

> Same environment across platforms (local/remote)
  - Software - CVMFS

> Graphical Jupyter extensions developed
  - Spark Connector
  - Spark Monitor
Configurable software environment - Prototype

> Adding support for Conda environments
   - Linked to Projects
   - Sharable

> Easy installation of extra packages
   - Clone/import Projects and install the software automatically

> Still a proof of concept
   - Integration with EOS is starting
SWAN Users’ Workshop

> First get together with the users of the service
  - Share use cases and knowledge among the community
  - Allow SWAN admins to understand how the service is used
  - Collect wishes for future improvements

> https://indico.cern.ch/event/834069
SWAN usage at COMPASS

For all the events candidates, extract the DVCS exact topology

\[ \Delta p = p_{\text{cam}} - p_{\text{spec}} \]

\[ \Delta Z_A = Z_A^{\text{cam}} - Z_A^{\text{ref and vertex}} \]

\[ M_{\text{Undet}}^2 = (k+p-k'-q-p)^2 \]

Pros & cons of SWAN

• Well organised analysis in notebooks using sections (use of markdown, latex, etc)

• Flexible enough to make systematic studies: use SWAN to adjust some cuts

• Drawback of interactivity: can be slow with too many events

Spark helps!

10/10/2019

Brian Ventura
Experiment namespace integration

Rucio & SWAN Integration idea – Mario Lassnig
ScienceBox: SWAN on Premises

> Packaged deployment of SWAN
- Includes all SWAN components: CERNBox/EOS, CVMFS, JupyterHub
- Deployable through Kubernetes or docker-compose

One-Click Demo Deployment
- Single-box installation
- Download and run in 5 minutes
  [https://github.com/cernbox/uboxed](https://github.com/cernbox/uboxed)

Production-ready Deployment
- Scale out service capacity
- Tolerant to node failures
  [https://github.com/cernbox/kuboxed](https://github.com/cernbox/kuboxed)
> SWAN is a CERN service that provides Jupyter Notebooks on demand
  - Promotes a cloud-based analysis model

> Valued by the community
  - Used for many use cases: Data analysis (Physics or others), Exploration, Teaching, …

> SWAN became a fundamental Interface for Mass Processing Resources
  - Currently it gives access to Spark
  - In development the access to GPUs and others