Future Generation Computer Systems 78 (2018) 1071–1078

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

SWAN: A service for interactive analysis in the cloud

Danilo Piparo *, Enric Tejedor, Pere Mato, Luca Mascetti, Jakub Moscicki, Massimo Lamanna

CERN, CH-1211 Geneva 23, Switzerland

HIGHLIGHTS

- A new service for web-based data analysis in the cloud is proposed: SWAN.
- SWAN combines CERN IT services with modern technologies of distributed computing.
- Synchronised cloud storage is a cornerstone of a data analysis service.
- SWAN users can synchronise their cloud storage space in their local machines.
- A cloud directory is the sharing unit of scientific results: code, text and data.

ARTICLE INFO

Article history:
Received 1 April 2016
Received in revised form
27 October 2016
Accepted 30 November 2016
Available online 6 December 2016

Keywords:
File sharing
Storage synchronisation
Cloud
Analysis
Mining
Notebook

ABSTRACT

SWAN (Service for Web based ANalysis) is a platform to perform interactive data analysis in the cloud. SWAN allows users to write and run their data analyses with only a web browser, leveraging on the widely-adopted Jupyter notebook interface. The user code, executions and data live entirely in the cloud. SWAN makes it easier to produce and share results and scientific code, access scientific software, produce tutorials and demonstrations as well as preserve analyses. Furthermore, it is also a powerful tool for non-scientific data analytics.

This paper describes how a pilot of the SWAN service was implemented and deployed at CERN. Its backend combines state-of-the-art software technologies with a set of existing IT services such as user authentication, virtual computing infrastructure, mass storage, file synchronisation and sharing, specialised clusters and batch systems.

The added value of this combination of services is discussed, with special focus on the opportunities offered by the CERNBox service and its massive storage backend, EOS. In particular, it is described how a cloud-based analysis model benefits from synchronised storage and sharing capabilities.

1. Introduction

For several years, High Energy Physics (HEP) has been facing unprecedented challenges in data storage, processing and analysis. As an example, the Large Hadron Collider (LHC) experiments at CERN [1] generate about 40 terabytes/s of raw data, which, after processing and filtering, results in tens of petabytes per year. During the last decade, the Worldwide LHC Computing Grid (WLCG) [2] has provided the infrastructure to store, distribute and analyse all this data. Scientists from around the world submit their jobs to the WLCG grid resources to execute their analyses on a daily basis.

Nevertheless, HEP is not the only community that has to confront with the big data challenge. Other examples in science include astronomy [3] and bioinformatics [4]. Industry is clearly leading the way in the field, especially big companies like Google, Amazon or Facebook, which mine customers' data for sales and marketing purposes [5]. Smaller-size organisations also have the means to collect and analyse fairly big amounts of data, mainly thanks to open source tools like Hadoop [6].

Among the directions explored by those communities, there is a noticeable trend towards *web-based interactive analysis*, where the user interacts with an on-line service by means of a web-browser [7–10]. This "software as a service" provisioning model allows users to focus on the solution of a problem in question rather than on installation, configuration and operational matters. Furthermore, such services are often backed up by computing and data resources that are hosted "in the cloud".

* Corresponding author.
E-mail address: danilo.piparo@cern.ch (D. Piparo).

http://dx.doi.org/10.1016/j.future.2016.11.035
0167-739X/

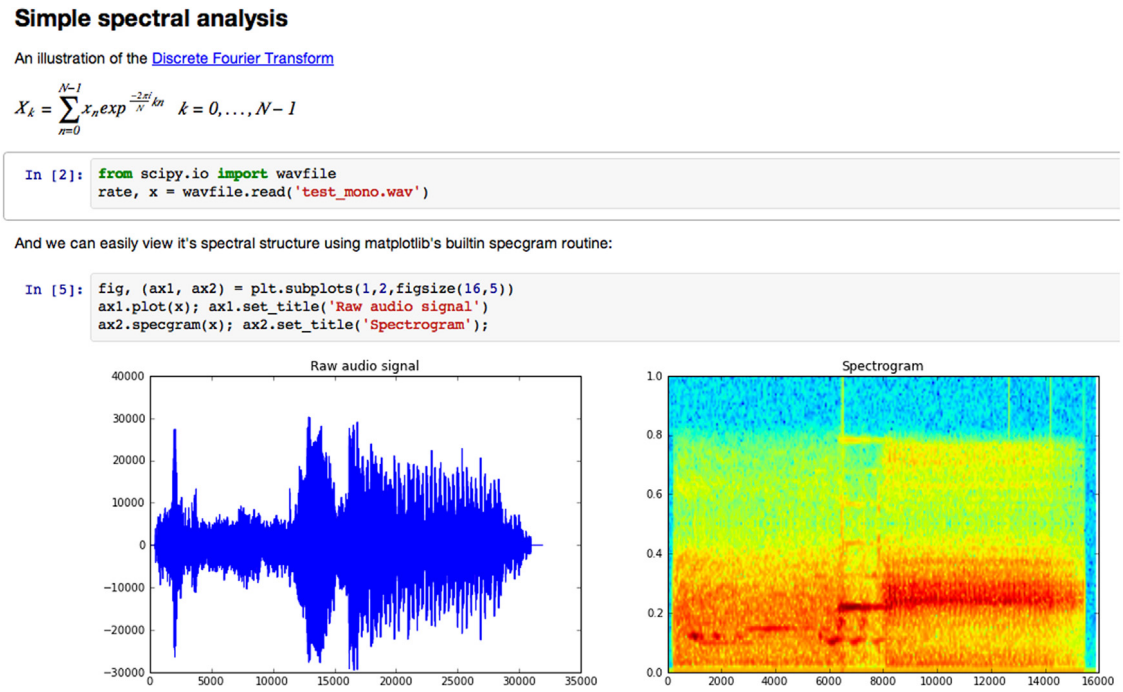

Fig. 1. An example of a Jupyter notebook [14]: text, formulae, code and images are combined in a computational narrative. Interactive JavaScript based widgets can also be used to provide increased interactivity and multimedia approach, e.g. for playing sounds or display videos [15].

These circumstances led to rethinking the data analysis models at CERN, more precisely in two ways: (i) how physicists could benefit from a service for interactive data analysis in the cloud, with only a web browser, and (ii) how state-of-the-art tools and existing CERN technologies could be combined to implement such a service, making it possible to access computing and storage resources transparently and on demand.

In particular, the cloud storage component plays a key role in the service and should fulfil three basic requirements: be the reference backend for both end-user and experiment data, be responsive enough to provide a good interactive user experience and provide easy means for scientists to synchronise their local workspaces and share their analyses.

In that sense, this paper presents the Service for Web based data ANalysis (SWAN), a cloud-based and interactive data analysis platform accessible via a web interface. SWAN boosts the productivity of scientists and engineers by allowing them to focus solely on the solution of their problems without investing resources in the creation, configuration and maintenance of software and hardware environments. Moreover, it facilitates the sharing of results and code, the access to scientific software, the achievement of reproducible results, the creation of tutorials, demonstrations for outreach and teaching, providing also many necessary elements for the preservation of data analysis procedures.

This paper is structured as follows. Section 2 introduces the interface that was chosen for SWAN, mainly based on the Jupyter [11] notebook platform for interactive data analysis. Section 3 describes the implementation of the service backend, i.e. how production-grade CERN IT technologies can be orchestrated in combination with other cutting-edge tools to build a distributed computing infrastructure. In Section 4 the role of the storage component in the service is characterised in detail. Section 5 is dedicated to the review of a series of use-case categories which are targeted by SWAN. Section 6 illustrates the main features of other work related to SWAN. Finally, Section 7 discusses the conclusions and Section 8 future work.

2. The notebook interface for interactive analysis

A common approach for interactive data analysis is to combine code, text, plots and rich media in the same document, known as *notebook*. Notebooks are divided in cells where the user can type code, execute it and see the results inline. In short, they can be thought of as an interactive programming shell running in a web browser.

There exist several notebook flavours [8,12,13], although one of them has been particularly successful: the Jupyter open source project [11]. Jupyter notebooks are an agile tool for both exploratory computation and data mining, and provide a platform to support reproducible research, since all inputs and outputs may be stored in a one-to-one way in the same document. Fig. 1 illustrates an example of a Jupyter notebook document where markdown text, formulae, code and plots are combined.

On the other hand, Jupyter is not restricted to a particular programming language, but instead it allows to plug in language extensions known as kernels. At the time of writing, more than forty programming languages are supported. Moreover, Jupyter can accommodate various ecosystems of tools for data analysis, e.g. R [16], Numerical Python [17] or Pandas [18].

The aforementioned attractive features of the Jupyter notebooks motivated their choice as the main interface of the SWAN service. Thus, users are able to produce their analyses in the form of notebooks by using only a web browser. The execution of the notebook cells, as well as the management of their associated data, happens seamlessly and transparently in the cloud. In order to support this cloud execution model of user notebooks, a service backend is needed; the details of such implementation are discussed next in Section 3.

3. SWAN and the portfolio of CERN services

The web-based interface of SWAN, based on Jupyter notebooks, is powered by a service backend that manages the execution of the notebooks on behalf of the users. Fig. 2 depicts the design of the SWAN backend, which strongly relies on a portfolio of already

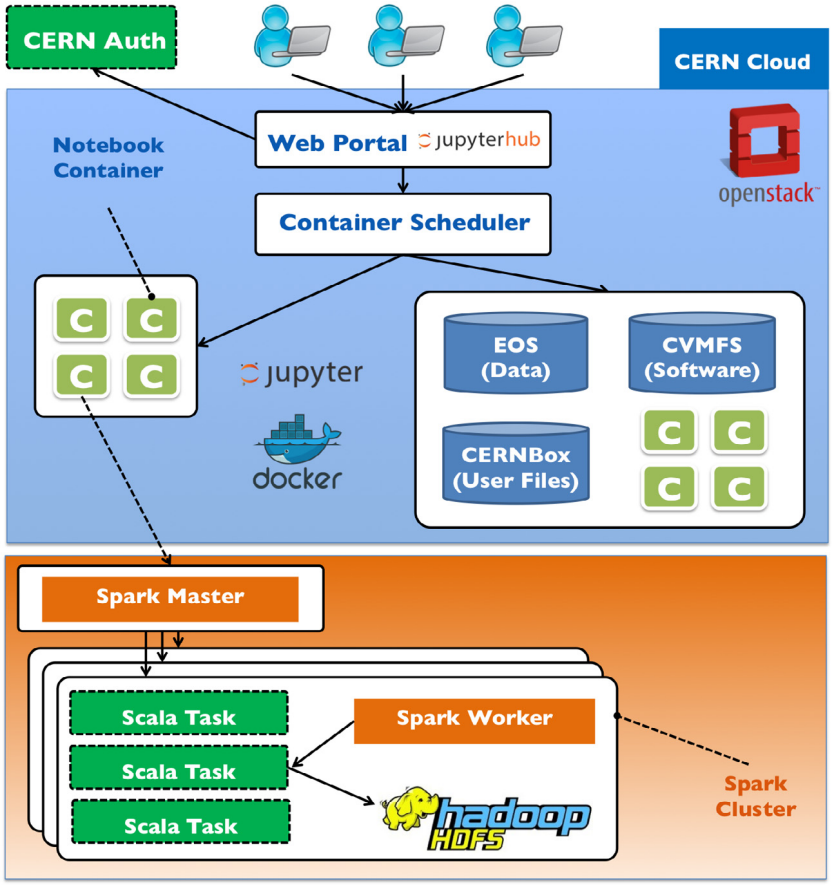

Fig. 2. In SWAN, users log in with their CERN credentials and request the creation of their notebook server through the JupyterHub [19] portal. The execution of notebooks is encapsulated in Docker containers, scheduled on the OpenStack infrastructure. The individual containers are lightweight: the needed software is available on CVMFS. Access to the CERN mass storage EOS is available, and the user's home directory in each container is the EOS portion private to each user synchronised via CERNBox (see Section 3.7). External resources, for example clusters, can be attached to the container if requested.

existing CERN services, leveraging as well some external cutting-edge technologies.

The next subsections will describe the SWAN building blocks with respect to authentication, user interface, computing infrastructure, software distribution and storage. The added value of SWAN is that, by combining all these pieces in a service, HEP scientists can access a complete analysis environment where they can work with technologies that are familiar to them. Thanks to SWAN, they are freed from the burden of locally installing and coordinating these technologies, being able instead to perform their analysis in the cloud with HEP software and accessing their data on CERN storage. Furthermore, they can easily share their work with other colleagues, as will be explained in more detail in Section 4.

3.1. Authentication and security

Upon connection to the SWAN service, the user's identity is checked via the single sign-on procedure of CERN. It is enough to type user name and password, which are validated through the CERN identity provider: this approach was preferred to a certificate-only authentication procedure for its simplicity.

The authentication process grants the user a personal notebook server in the cloud to execute analyses, as well as access to cloud storage and batch resources according to the user's credentials.

The single sign-on login procedure is only one of the several security measures which were adopted for SWAN. The full palette of requirements dictated by CERN security were satisfied, ranging from the operating system versions of the host servers to Docker container [20] settings to properly encapsulate user sessions. The use of the Docker technology will be further explained in Section 3.4.

3.2. Web portal

Once authenticated, the user is redirected to a web portal that is based on the JupyterHub [19] technology. The main function of JupyterHub is to manage the login of single users and to dispatch their sessions on the dedicated computing resources in a well encapsulated environment (see Section 3.3).

In the JupyterHub portal, the user can request the creation of a personal notebook server (see Fig. 3).

Furthermore, the server can be customised with a web form that contains two types of items:

- *Software environment*: the user is able to select any software collection (libraries, packages) required for the execution of the notebooks (Fig. 4). In a first implementation, this corresponds to pre-defined groups of packages offered by the service. The possibility of allowing the user to setup custom environments will be also granted in the future.
- *Hardware resources*: in addition to the resource where the notebook server runs, SWAN will allow to customise hardware resources that can provide more computing power or, more in general, performance. Examples are the access to an external Spark [21] cluster where computations are spawned and parallelised, a multicore node or a machine equipped with an accelerator.

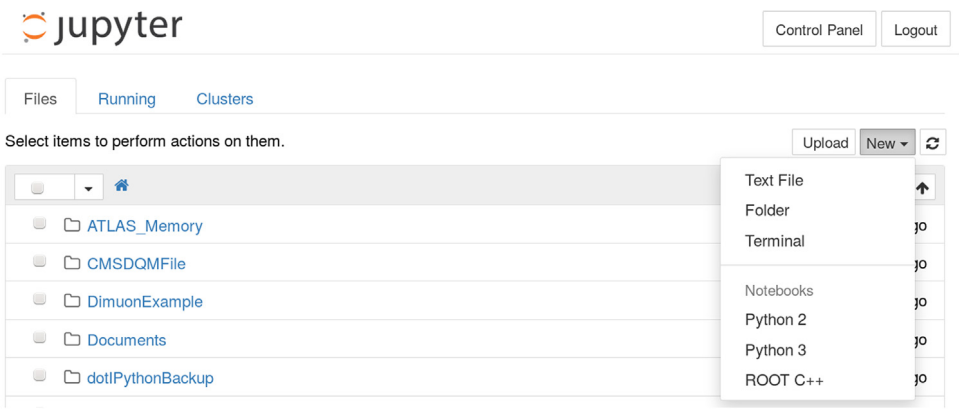

Fig. 3. The web interface of the Jupyter server. The content of the home directory is displayed. The drop-down menu shows that it is possible to create a text file, a directory or open a terminal. Different notebook flavours can be seen too. In this example Python2, Python3 and C++ languages are available. The interface also allows to upload files.

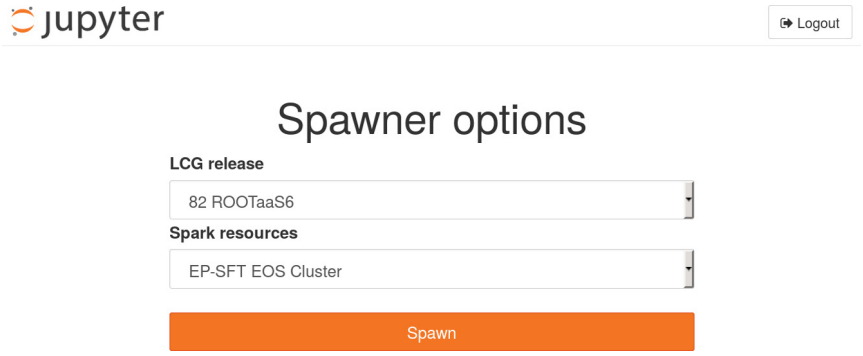

Fig. 4. A simple web form allows the user to select a collection of software packages available from within the container (LCG release). In the future, it will be also possible to specify hardware resources such as Spark clusters, multicore nodes, GPGPUs or accelerators.

3.3. Virtualised infrastructure

The SWAN service is able to leverage the CERN OpenStack [22] cloud instance for hosting purposes. This means that both the web portal and the user notebook servers can run in a virtualised environment, which is in charge of executing the calculations that are described in the notebooks.

As a result of running in the cloud, the number of resources dedicated to SWAN can be elastically increased or decreased depending on the workload (e.g. amount of users that are actually logged in), thus avoiding under or over-provisioning.

Regarding resource accounting, the service entirely relies on the policies enforced by the CERN cloud managers.

3.4. Containers

When a user requests the creation of their personal notebook server, a Docker container is allocated to encapsulate that server. Hence, all the calculations that a particular user executes in a notebook are executed inside a container. The operating system of the container image chosen for the first deployment is Scientific CERN Linux 6 [23] (SLC6). At the time of writing, the vast majority of CERN machines and computing nodes of the WLCG are equipped with the SLC6 operating system.

Containers feature adequate characteristics for the SWAN use-case:

- *Isolation*: even if they run in the same virtual resource, every container is isolated from the other containers and from the host. This approach is transparent to the user and is beneficial both in terms of fairness (a process in a container cannot affect processes running in other containers) and security (the harm caused by a malicious user is restricted to the scope of the container).
- *Runtime and responsiveness*: a container is much faster to spawn than a full virtual machine on equivalent hardware. This advantage is exploited to improve the user experience when requesting the creation of a container.
- *Ease of deployment*: SWAN relies on a single Docker image with the software that is strictly necessary to run the notebook server, which can then be deployed on the nodes that host the containers. There is no need to provide a full image per use case thanks to lightweight contextualisation, a concept discussed in detail in 3.5.

3.5. Software distribution

A distinctive feature of the SWAN service is that the analysis software is made available via a distributed file system, CVMFS [24]. CVMFS is the CERNVM File System: an aggressively cached, HTTP-based file system designed for the distribution of software. Since files are lazily cached once accessed, CVMFS is mounted on the virtual machines of the SWAN infrastructure and mounted as volume inside containers. Containers running on the same virtual host share a common cache therewith increasing the probability of a cache hit. CVMFS can potentially store any required software, including (but not restricted to) LHC experiments' software stacks.

A single Docker image is used to run the notebook servers and is lightweight: it only contains an installation of Jupyter, whereas CVMFS is in charge of provisioning the rest of the software.

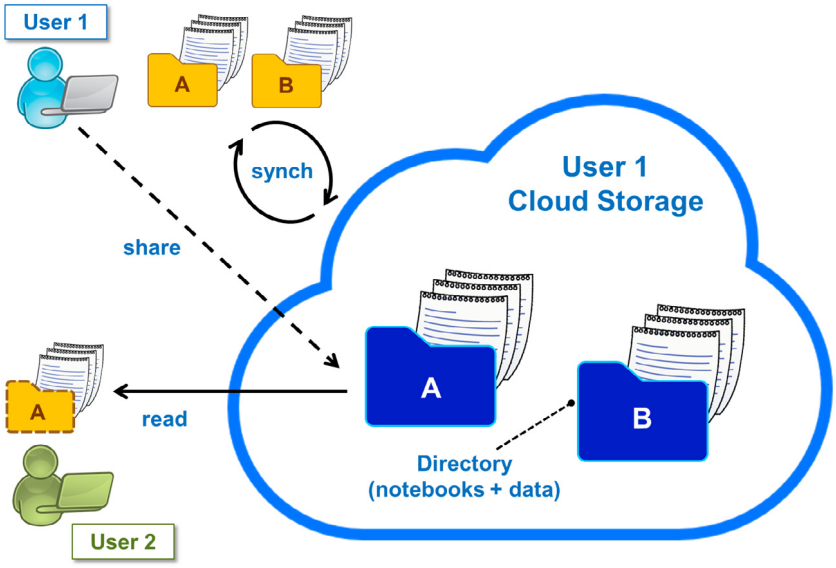

Fig. 5. User 1 is synchronising with CERNBox two directories in the cloud. They contain notebooks and other data files. Directory A can be shared with User 2.

This approach allows a veritable *lightweight container contextualisation* which has at least three desirable features:

- allows the support of several different analysis software environments,
- makes it possible to upgrade of the set of packages available from within the container without interfering with the container image,
- avoids network congestion in container environments [25] due to the usage of large images.

As described in Section 3.2, a user can customise the software environment by means of a web form, right before spawning the notebook server. This form is linked to CVMFS, since it allows to select a given group of software configurations to contextualise the container.

3.6. Analysis software

Thanks to CVMFS, SWAN is able to access any required software, including libraries for carrying out data analysis. The aim of SWAN is not to be restricted to any analysis ecosystem in particular, but rather to give the user the freedom to choose and to exploit the synergies among them, also through the combination of different programming languages. Examples of such ecosystems are R, Pandas and also ROOT [26], a data analysis framework widely adopted in HEP.

3.7. Storage

One of the most important components of the SWAN service is cloud storage. The choice of technology for SWAN to implement such storage is EOS [27]. In SWAN, the EOS namespace is visible from within every container via a fuse-mount procedure which preserves the credentials of the user. EOS is available on the host server and mounted as a data volume on the containers encapsulating users sessions.

As the mass storage solution for CERN, EOS already stores data of the LHC experiments and their simulations as well as non-physics data such as logs coming from the computing infrastructure monitoring.

On the other hand, CERN makes available to every user a private and synchronised portion of EOS, called CERNBox [28]. Section 4 is dedicated to the description of this type of storage.

4. The role of synchronisation and sharing

CERNBox is the CERN storage technology for file synchronisation and sharing, based on ownCloud [29]. It features clients to synchronise one or more local directories into the cloud, which can then be synchronised back into any device (desktop machines, tablets, phones, etc.) or shared with other users (see Fig. 5).

4.1. Synchronisation

In SWAN, users see the content of their CERNBox as home directory. This means that all results, datasets, documents (or more generally, files) produced in a SWAN session are located in a synchronised storage area. Consequently, all these files will also be visible from any client and vice versa: files edited locally will appear on the SWAN server side. This allows to complement the analyses made online also offline, de facto constructing a cross-platform transport of the analysis procedures, together with their input and output datasets. Therefore, in the context of SWAN, synchronised storage becomes the gateway to the cloud.

One of the consequences of the feature just described is that the user can transparently mirror the local environment in the cloud. This is an innovative aspect — none of the commercial and open source cloud analysis providers mentioned in Section 1 features this characteristic; instead, a clear separation is present between local user and cloud storage. Furthermore, this solution represents a paradigm shift at CERN from the, at the time of writing, adopted distributed file system mounted on all Linux machines of the Organisation, AFS [30].

Finally, this mechanism also allows scientists to customise the environment for their analysis, for example making available to notebook servers libraries and software packages of their choice.

4.2. Sharing

Notebooks only make sense in a context of sharing. People produce notebooks so that others can see or review their content, modify, re-run and, possibly, re-share them. Since notebooks are the main interface of SWAN, sharing is a central feature of the service.

For that purpose, we rely entirely on CERNBox and its sharing capabilities and, in particular, its integration with the

authentication system of CERN, which makes possible to select individuals or groups with which data can be securely shared. Concretely, sharing can be triggered via the CERNBox web interface, the unit of sharing being a directory. Such granularity fits very well the SWAN use case, since a scientific result is rarely represented by a single file, but rather by a coherent ensemble of text, code, input and output data.

Thanks to the sharing capabilities of CERNBox and its integration with the authentication systems of the Laboratory, progress and findings documented in notebooks can be debated more easily among colleagues and refined by collaborative efforts, for example to perform stringent cross-checks before a discovery.

In presence of SWAN, the sharing potential implied by federated clouds acquires a new type of added value. It becomes a procedure based on directories' sharing which allows to strengthen the collaboration among sciences, research laboratories and universities.

5. Summary of targeted use cases

This section characterises some representative use cases where the SWAN platform has an impact, boosting the productivity of the researchers.

5.1. Scientific and non-scientific data analysis

SWAN gives CERN the opportunity to integrate the data analysis approaches of its scientists and engineers. It provides all the building blocks both for particle collisions studies and analytics of information coming from monitoring of computer farms or electronics equipments. By utilising the same infrastructure, the two communities can build a common language and analysis procedures. The ability to seamlessly, selectively and securely share results, code and data not only boosts the performance of users but also gives the opportunity to rethink their approaches taking inspiration from a different mindset.

5.2. Tutorials and teaching

SWAN simplifies the setup of tutorials. One of the biggest challenges posed by teaching is the need to provide a uniform environment to all the participants in order to offer fruitful hands-on sessions. It is not always possible to provide a centrally-managed terminal to everybody. In addition, even assuming that all participants have a laptop at disposal, the distribution of a virtual machine is also inconvenient because of the size of the image and ultimately the virtualisation support of the underlying hardware. The usage of SWAN solves all the aforementioned problems providing a uniform environment accessible with a web browser from any device. The presence of synchronised storage allows the audience of the lectures to keep a copy of the work performed during the hands-on sessions on their devices. The possibility of sharing results with tutors, teachers and peers is an opportunity to achieve an enriching learning experience.

Before building the final architecture illustrated in Fig. 2, several intermediate milestones providing a subset of the features of the final product were achieved. These prototype services were leveraged in several occasions such as analysis tutorials for undergraduate students at CERN [31], allowing not only to simplify teaching but also to collect feedback from early adopters.

5.3. Outreach and open data

SWAN is a platform which offers all the features to be an ideal support for the CERN Open Data [32] initiative. The open datasets are directly accessible via EOS. The notebook interface is suited for scientists and students with backgrounds also different from High Energy Physics. It gives access to several languages and scientific software packages. In addition it grants a way to unify in the same document all the explanations and examples which are needed to transform a prestigious discovery into a high-school or university lecture, immediately shareable world-wide.

5.4. Geographically remote collaborators

The success of an organisation like CERN heavily relies on world wide collaborations. For example, LHC experiments are sustained by collaborations of research and university institutes from all over the world. Presently it is very hard or impossible to exploit CERN interactive login services from institutes far from the Laboratory. For example, the network latencies make the usage of the interactive logon service to Linux for all CERN users, LXPLUS [33], very inefficient without a graphical connection, impossible when this feature is requested. In addition, a small bandwidth can be an obstacle for the transfer of produced results and datasets from and to CERN.

The Jupyter web interface is able to mitigate if not to entirely hide latencies thanks to an asynchronous web frontend. On the other hand, synchronised storage, in combination with sync-clients for laptops or mobile devices, greatly facilitates transfers in presence of little bandwidth.

6. Related work

Apart from SWAN, there exist other products that follow the trend of web-based interactive analysis hosted in the cloud. Most of them are commercial solutions [7,34,8,9], some of which are offered by big companies [35,10,36]. Unlike these approaches, SWAN is a non-profit service to support the data analysis workflows of CERN employees and collaborators. Furthermore, a few of the aforementioned solutions can be deployed on premises, whereas new instances of SWAN could be hosted by other centres using their own infrastructure. Regarding cloud storage, only SWAN amongst all these products can mirror the cloud user space on a local user machine (and vice versa), thus making possible to easily exchange data with the cloud via CERNBox. Finally, none of the discussed solutions offer complete freedom in the distribution of software, for example tools widely adopted in our community such as ROOT are not available; this prevents the possibility to read the data the LHC experiments acquired.

An exception to the products cited above is Binder [37], which is a free online service that offers notebooks on demand. Binder presents the following limitations that do not exist in SWAN: first, the computing power pledged is just enough to share demonstrators; second, the storage is volatile: nothing like a users' space is present; lastly, the software needed by the application must be included in the Docker container used to isolate the user sessions, leading to extremely large images (i.e. tens of Gigabytes) which are not easily manageable on commodity clusters.

7. Conclusions

SWAN is a service for interactive data analysis in the cloud, developed at CERN. It combines leading edge technologies such as Jupyter notebooks and Docker with a portfolio of existing CERN IT services, namely virtualised cloud resources, user authentication, specialised clusters and storage.

Cloud storage plays a key role in SWAN. Both experiments' data and private user files live in the cloud, so that user analyses can access them from within the service. In addition, thanks to CERNBox, users can choose to synchronise their private space on local resources, thus creating a gateway to the cloud. This feature

represents an innovation with respect to similar services, which do not allow clients to work locally with cloud data. Finally, CERNBox also makes possible to share data among users at directory level, which matches the granularity of an analysis (e.g. a notebook plus some input/output data).

Another innovative feature of SWAN is the combination of Docker containers and CVMFS for software distribution. This solution is based on having a single, thin container image and decoupling software provisioning from it; instead, a lightweight contextualisation is performed through a distributed file system.

8. Future work

A rich program of developments is foreseen for SWAN. Keeping up with the highest security standards will be an activity with high priority for the project. Injecting data related to scientific activities in third party Clouds can rise security concerns. This is one of the reasons why an effort for improving the current packaging of SWAN will take place in order to make the service deployable with less effort on sites other than CERN, possibly relying on technologies different from the ones described and allowing universities, laboratories and companies to keep their data on-premises. The container scheduler will be able, when needed, to start containers on hosts featuring special resources, for instance hardware accelerators or other types of special resources. The web interface of SWAN will benefit from many ameliorations and customisations aiming to increase its usability. In addition, the service will evolve in order to accommodate the execution of applications different from Jupyter notebooks, see for example [38].

References

[1] European Organisation for Nuclear Research, http://www.cern.ch.
[2] Worldwide LHC Computing Grid, http://wlcg.web.cern.ch.
[3] The Square Kilometre Array (SKA Telescope), http://www.skatelescope.org.
[4] EMBL — European Bioinformatics Institute, http://www.ebi.ac.uk.
[5] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, J.Q.n. Candela, Practical lessons from predicting clicks on Ads at Facebook, in: Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, ADKDD'14, ACM, New York, NY, USA, 2014, pp. 5:1–5:9. http://dx.doi.org/10.1145/2648584.2648589, URL http://doi.acm.org/10.1145/2648584.2648589.
[6] T. White, Hadoop: The Definitive Guide, O'Reilly Media, Inc., 2012.
[7] Wakari: Web-based Python Data Analysis, http://wakari.io.
[8] SageMathCloud: Collaborative Computational Mathematics, http://cloud.sagemath.com.
[9] Plotly: Make charts and dashboards online, http://plot.ly.
[10] Microsoft Azure Machine Learning, http://studio.azureml.net.
[11] Jupyter: Open source, interactive data science and scientific computing, http://jupyter.org.
[12] Wolfram Notebook Technology, http://www.wolfram.com/technologies/nb.
[13] Apache Zeppelin, http://zeppelin.incubator.apache.org.
[14] The IPython Notebook, http://ipython.org/notebook.html.
[15] IPython widgets, http://ipywidgets.readthedocs.org/en/latest.
[16] The R Project for Statistical Computing, http://www.r-project.org.
[17] P.F. Dubois, Extending Python with Fortran, Comput. Sci. Eng. 1 (5) (1999) 66–73.
[18] Python Data Analysis Library, http://pandas.pydata.org.
[19] JupyterHub: A multi-user server for Jupyter notebooks, http://github.com/jupyter/jupyterhub.
[20] Docker: build, ship and run any app, anywhere, http://www.docker.com.
[21] M. Zaharia, M. Chowdhury, M.J. Franklin, S. Shenker, I. Stoica, Spark: Cluster computing with working sets, in: Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing, HotCloud'10, USENIX Association, Berkeley, CA, USA, 2010, p. 10. URL http://dl.acm.org/citation.cfm?id=1863103.1863113.
[22] OpenStack: Open Source Cloud Computing Software, http://www.openstack.org.
[23] Scientific Linux CERN 6, http://linux.web.cern.ch/linux/scientific6.
[24] J. Blomer, C. Aguado-Sánchez, P. Buncic, A. Harutyunyan, Distributing LHC application software and conditions databases using the CernVM file system, J. Phys. Conf. Ser. 331 (4) (2011) 042003. URL http://stacks.iop.org/1742-6596/331/i=4/a=042003.
[25] We are working with CERN to tackle network congestion in container environments (blog post), https://mesosphere.com/blog/2016/03/08/cernvmfs-mesos-containers.
[26] R. Brun, F. Rademakers, ROOT — An object oriented data analysis framework, Nucl. Instrum. Methods Phys. Res. A 389 (1–2) (1997) 81–86. http://dx.doi.org/10.1016/S0168-9002(97)00048-X, New Computing Techniques in Physics Research V, URL http://www.sciencedirect.com/science/article/pii/S016890029700048X.
[27] A. Peters, E. Sindrilaru, G. Adde, EOS as the present and future solution for data storage at CERN, J. Phys. Conf. Ser. 664 (4) (2015) 042042. URL http://stacks.iop.org/1742-6596/664/i=4/a=042042.
[28] L. Mascetti, H.G. Labrador, M. Lamanna, J. Moscicki, A. Peters, CERNBox + EOS: end-user storage for science, J. Phys. Conf. Ser. 664 (6) (2015) 062037. URL http://stacks.iop.org/1742-6596/664/i=6/a=062037.
[29] ownCloud, https://owncloud.org.
[30] J.H. Howard, M.L. Kazar, S.G. Menees, D.A. Nichols, M. Satyanarayanan, R.N. Sidebotham, M.J. West, Scale and performance in a distributed file system, ACM Trans. Comput. Syst. 6 (1) (1988) 51–81. http://dx.doi.org/10.1145/35037.35059, URL http://doi.acm.org/10.1145/35037.35059.
[31] ROOT Summer Students Tutorial 2015, https://indico.cern.ch/event/407519.
[32] CERN Open Data Portal, http://opendata.cern.ch.
[33] The LXPLUS Service, http://information-technology.web.cern.ch/fr/services/lxplus-service.
[34] Wolfram Mathematica Online, http://www.wolfram.com/mathematica/online/.
[35] IBM Bluemix, https://www.ibm.com/cloud-computing/bluemix/.
[36] Google Cloud Datalab, https://cloud.google.com/datalab/.
[37] Binder, http://mybinder.org.
[38] Akhet, https://github.com/AkhetLab.

Danilo Piparo is a physicist, software and service architect at CERN, Geneva, Switzerland.

He obtained a Ph.D. in Physics at the Karlsruhe Institute of Technology (KIT). He was involved in the Higgs discovery working for the CMS experiment.

As an expert of parallel software development and performance tuning, always focussed on improving the efficiency of data processing in scientific computing.

Presently he is responsible for the I/O subsystem of the ROOT package but he is actively involved in the development and support of reflection and mathematical libraries.

He is passionate about distributed systems such as grids and clouds.

Enric Tejedor received his Ph.D. from the Technical University of Catalonia (UPC, Spain) in 2013. He conducted his doctorate research as a member of the Grid Computing and Clusters group of the Barcelona Supercomputing Center, where he participated in several EU research projects.

As part of his Ph.D., he also carried out two internships at the IBM T.J. Watson Research Center (NY, USA). In 2015 he started working at CERN (Switzerland) as a senior software engineer. Enric is passionate about parallel programming and distributed systems such as HPC clusters, Grids and Clouds.

Pere Mato studied physics at University of Barcelona, Spain, where he obtained the Ph.D. in 1990. Since 1986 has been working at CERN in a number of projects. Started with the 3081/E emulator project at the DD division, and later moved to the Aleph experiment in the area of DAQ and slow controls. In 1994 he took the overall responsibility of the Aleph TPC detector until the end of LEP. Since 1998 he has been leading the development of the core software and framework for the LHCb experiment (Gaudi) and later the LCG Core Libraries and Services project (SEAL). In 2005 has been appointed Applications Area manager of the LCG project.

Luca Mascetti, engineer in the IT Data Storage group at CERN, member of the team in charge of operating and supporting multi-petabytes storage system services for CERN and the Worldwide LHC computing grid community (notably CERNBox, EOS, CASTOR, NFS and Ceph), ensuring dependable and secure operations of the critical storage services for the LHC scientific program.

In charge of innovative strategic storage services, operating different technologies developed at CERN and in the open source community. In this role I am in charge of the communication for our storage services across the different communities in IT and at CERN, as well as with our partners outside the Organisation and I support a very large user community.

In 2011 I started my career at CERN with the Graduate Engineering Training programme (GET) after receiving a Master degree in Computer System Engineering in December 2010 at Politecnico di Milano, Italy with a thesis concerning the Green Traffic engineering where I developed optimisation methods for energy savings in IP traffic engineering.

Jakub T. Mościcki is a researcher, software engineer and service manager at CERN, Geneva, Switzerland. He obtained the Ph.D. in Computer Science from the University of Amsterdam (UvA). He was involved in designing and developing frameworks for supporting distributed computing for the LHC experiments as well for multidisciplinary applications in theoretical physics, medical and radiation studies, bio-informatics, drug design and telecommunications.

He now works in the Data Storage group which designs and operates large-scale disk and tape storage for LHC physics data as well as the general purpose file system and Cloud Storage services.

Massimo Lamanna received a Ph.D. in High-Energy Physics in 1993 at the Trieste University, Italy.

He has over 15 years of experience in the field of scientific computing leading projects in the area of data management, monitoring, user access to grid resources and user/community support within the High-Energy Physics community. Notably he coordinated different activities in the area of distributed computing in the LCG project (LHC experiments) and in the COMPASS and ATLAS experiments.

He also fostered collaboration across user communities (Biology, Climatology, Telecommunication, etc.) and initiated the EGEE User Forum, one of the largest and most active events in the grid computing field.

For the last 4 years he is responsible for all the disk-data management operations at CERN. He is responsible for managing the data from the LHC experiments to the CERN computer centres; for the CERN disk farms exchanges with collaborating centres world-wide; to serve data to be processed and recorded to tape; to enable thousands of physicists for the final data analysis.