ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA057

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

BUILDING A DATA ANALYSIS AS A SERVICE PORTAL*

A. Campbell†, A. Götz, European Synchrotron Radiation Facility, Grenoble, France
D. Salvat, A. Camps, ALBA Synchrotron, Cerdanyola del Vallès, Spain
G. Kourousias, I. Andrian, Elettra Sincrotrone Trieste, Trieste, Italy
M. van Daalen, S. Egli, Paul Scherrer Institut, 5232 Villigen PSI, Switzerland
on behalf of the CALIPSOplus JRA2 project

Abstract

As more and more scientific data is stored at photon sources as part of implementing FAIR data policies there is a growing need to provide services to access to view, reduce and analyse the data remotely. The CALIPSOplus project, in which all photon sources in Europe are involved in, has recognised this need and created a prototype portal for Data Analysis as a Service. This paper will present the technology choices, the architecture of the blueprint, the prototype services and the objectives of the production version planned in the medium term. The paper will cover the challenges of building a data analysis portal from scratch which covers the needs of multiple sites, each with their own data catalogue, local computing infrastructure and different workflows. User authentication and management are essential to creating a useful but sustainable service.

INTRODUCTION

Photon sources are hitting a data analysis wall with the huge increase in data volumes, new techniques and new user communities. Users are limited by the difficulty in exporting huge data from the source to their home labs and by the lack of easy access to data analysis programs. At the same time science is becoming more open by sharing the data, methods and publications - the so-called **O**pen Science movement. Making the data used in publications easily available enables others to reproduce the analysis and findings. This has been one of motivations for photon sources to adopt open data policies. Another motivation has been the need to alleviate the data analysis bottleneck by implementing modern data management so that data analysis services can be built on top of data catalogues. Lack of adequate data management limits the data services which can be be offered to users. In order to address these issues it was decided to include data management as a special topic in H2020 project CALIPSOplus [1].

CALIPSOplus is a project of all synchrotrons and free electron lasers in Europe and the Middle East to contribute to the completion of the European Research Area [2] and to tackle the new challenges arising from the commitment of the EU to Open science, Open innovation and Open to the world. The project aims at pushing integration with respect to user access, support, instrumentation, data analysis and data management forward. The goal is to harmonise data management and data analysis services of the different facilities. It was therefore agreed to start a Joint Research Activity (JRA) to prototype data analysis services on top of the data portals which are part of the data policies being implemented at photon sources in Europe. This paper describes the CALIPSOplus prototype data analysis portal.

DATA ANALYSIS AS A SERVICE

The *as a Service* (aaS) family of services include the well known *Software as a Service* (SaaS) and *Data as a Service* (DaaS) services. In order to address the data analysis bottleneck problem we propose to bring the data analysis software to the data and provide *Data Analysis as a Service* (DAAS). Data analysis can include providing data reduction services to calibrate and reduce the data to smaller volumes or different dimension e.g. from raw detector counts to physical units. The goal of DAAS is to leave data at the source and provide users with remote access to data analysis programs and algorithms. The DAAS prototype is aimed at identifying and implementing a number of prototype services for a selected set of use cases. The experience gained and feedback from users will help move towards a production ready service for data analysis in the future.

USE CASES

A survey was performed among the different CALIPSOplus members to compile a list with reusable scientific analysis software use case candidates at the different facilities. Feasible candidates for use cases were the applications for online analysis of scientific data, which are at the "end" of a data analysis chain, and therefore producing results which are directly useful for publications. However, any other application considered re-usable from other sites could be nominated. "Application" could also mean libraries, toolboxes or components from which full applications could be more easily derived. The following software packages were selected as use cases, see deliverable [3] and the software catalogue [4] for the details about the software and required dependencies:

- **CrystFEL** (DESY) : CrystFEL is a suite of programs for processing diffraction data acquired serially in a snapshot manner, such as when using the technique of Serial Femtosecond Crystallography (SFX) with a free-electron laser source.

* This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No 730872.
† contact aidan.campbell@esrf.fr

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA057

- **Ptycho Shelves** (PSI): Ptycho Shelves is an innovative and conceptually-simple modular framework for ptychography reconstructions.

- **Savu** (Diamond) : Savu is a Python package to assist with the processing and reconstruction of parallel-beam tomography data.

- **pyFAI** (ESRF) : the pyFAI library reduces 2D images into 1D curves (azimuthal integration). As a library, the aim of pyFAI is to be integrated into other tools or being used directly within Jupyter notebooks.

- **PyMca** (ESRF) : PyMca allows to explore X-ray microscopy datasets.

- **Crispy** (ESRF) : Crispy is a modern graphical user interface to simulate core-level spectra using semi-empirical multiplet approaches.

- **PyNX** (ESRF) : Python toolkit for accelerated Nano-structures Crystallography and Coherent X-ray Imaging techniques.

SOFTWARE CATALOGUE

Almost all the use cases listed above are in the PaNData software catalogue. The PaNData software catalogue [4] is a database of software used mainly for data analysis of photon and neutron experiments. It allows scientific institutes to upload information about the software that they have created for others to download although the website does not contain the binaries for the software.

All software developed for the implementation of the proposed architecture is being published on Github under permissible licenses at `https://github.com/calipsoplus`

BLUEPRINT

At present, we are working to provide five main services which users can access remotely to perform their data analysis linked to their data, these include:

- Data browsing
- Jupyter Notebooks [5]
- Virtual machines and containers
- Remote desktop applications
- Web applications for data analysis

The blueprint (see Fig. 1) was prepared at a meeting in May 2018 where the collaborators met to discuss requirements for the blueprint and to design the potential architecture and implementation of these services. The blueprint design was completed at a hands-on meeting held at the ALBA synchrotron in October 2018.

It was agreed that applications will be implemented as deployable packages, as pre-configured virtual machines or as containers. Virtual machines and containers provide encapsulated user environments, which can eventually be archived with the experimental data, thereby capturing valuable provenance data and strongly supporting the reproducibility of the original experiment and data analysis workflows.

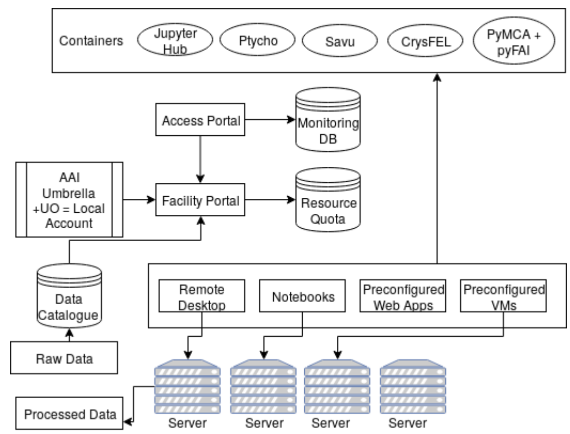

Figure 1: Blueprint of CALIPSOplus.

Infrastructure Requirements

The infrastructure on which the DAAS services will run must be adequately dimensioned for the service to enable users to work comfortably. Even if the blueprint could run on a single computer it will require one or more clusters of machines to provide a useful service. The exact number of machines will depend on the use cases implemented at each site and the number of users.

Portals

Access Portal The current implementation has an access portal which can be viewed by anyone and can be used for both promoting the capabilities of each facility as well as providing access to the portal of each site. Each site has an authorisation database in order to prevent unauthorised access. CALIPSOplus has chosen UmbrellaID [6] as the main authentication mechanism. UmbrellaID allows anyone to make an account therefore users must be able to link their UmbrellaID account with their local site account to make sure that only authorised users (those who carried out an experiment at the facility) have access to the services offered by the DAAS portal. OpenID Connect (OIDC) [7] is supported to enable local access without an UmbrellaID account. This allows some facilities to make use of their existing Single Sign-On (SSO).

In the future, the CALIPSOplus DAAS portal could be a thematic service in the EOSC [8] for promoting common PaN services like the software catalogue, tutorials, documentation, open data etc.

Facility Portal The facility portal is the portal implemented by each synchrotron with bespoke services offered including the core functionality to be offered by all sites.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA057

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

The resource quota database is connected to the portal in order to make sure that users only make use of a number of resources proportional to their experiment/proposal. The importance of a resource quota is to make sure that a single user or a small group of users are not using all of the resources and preventing others from accessing the resources. Resource quotas allow control of the number of CPUs, RAM, storage and time allocated to each user via the DAAS portal. The quotas depend on what users need to do. A user carrying out computationally heavy simulations will require more resources and will need to accommodated for, but not every user will need this or be given additional resources. For JupyterHub, the initial plan is to give each user the same amount of resources e.g storage, RAM and CPU which they can use.

Data Catalogue The facilities' data catalogue is linked to the facility portal as the user will have to specify which dataset(s) they wish to have access to either on a virtual machine or for use on JupyterLab. The user will enter some details about the dataset such as the proposal number, the name of the beamline, session, date of experiment etc. based on which the data catalogue will be able to find the data and return it to the user in the most efficient way. This data will be mounted directly to the virtual machine they have access to or by the Jupyter Notebook container. This operation will remain the same for the user provided it is stored on a hard drive at the facility. Data that is archived is not restored automatically. It will require a manual request via the data portal to restore the data. The user will need to specify where the data has been restored to access it.

Orchestrator

The orchestrator coordinates containerised services started via the portal for multiple users. The orchestrator (Kubernetes and/or SLURM) will deploy the containers based on the availability of resources on each node as well as the hardware requirements that have been set for the container such as CPU, RAM and GPU.

Servers Each of these services will require servers in order to host them. In the case of Jupyter Notebooks, they will be hosted on a Kubernetes cluster which will allow load balancing of many containers across multiple servers. With this setup, it is possible to add and remove servers depending on the number of users and their requirements.

As most facilities are using SLURM, deploying notebooks using SLURM was implemented however this involves some restrictions as users are unable to install additional software and must rely completely on the software packages already installed on the node.

Each facility will have the choice to use Kubernetes [9], SLURM or both orchestrators to provide Jupyter Notebooks.

As part of the services supplied, it was possible to provide containers with pre-configured software such as CrystFEL [10] and Ptycho [11] installed. Docker containers were used however Singularity [12] is another possibility.

Container Orchestrator As not all facilities have a production ready Kubernetes deployed, they are still able to make use of Docker to provide containers with pre-installed software. This is used for simple applications as Docker is limited to a single machine without using Docker Swarm.

It will be necessary to take into account not only performance but also hardware requirements as virtual machines are much more resource intensive. It is possible that containers might be able to provide a sufficient performance with less resources.

IMPLEMENTATION

The project is currently in the implementation phase: creating an initial prototype. The prototype follows the original blueprint closely despite encountering some problems along the way. The difficulties experienced were in the following areas :

- Mounting NFS directories to containers with correct user permissions
- Accessing users experimental data from metadata catalogues

A Jupyterhub has been setup to run on Kubernetes which creates containers for each Jupyter Notebook. As these containers were run using root as default, users were unable to access their home directory and the experimental data on NFS due to permission errors. Working with some of the main contributers on Jupyterhub for Kubernetes it was possible to run the container as the user instead of root and have the NFS paths mounted at start-up.

One of the biggest problems remaining is accessing data from ICAT due to the fact that ICAT does not currently support OAuth2. This requires the user to login twice in order to get an authorisation token and access the data. This problem can affect facilities not using ICAT as well. Each facility will contribute code to support their data access platform.

Frontend

The frontend of the web application is being written in Angular 6 which supports multiple packages for OpenID Connect as well as to enable us to make use of software which has already been created to enable remote desktop applications via RDP and VNC. See Figures 2 and 3 for examples of the frontend service pages.

REST API

The backend of CALIPSOplus is written in Python making use of the Django [13] framework. Python was chosen as it provides many APIs which allows one to create containers and virtual machines using OpenStack, Docker and Kubernetes more easily. Django provided a rich ecosystem of third-party applications such as OpenID Connect for authentication, notification system for sending emails and out-of-the-box support for many types of databases giving us great flexibility.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA057

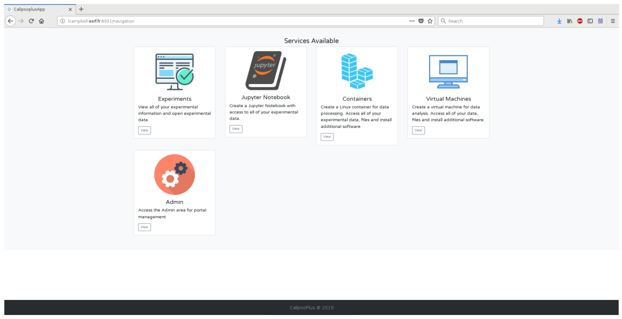

Figure 2: Front End services available.

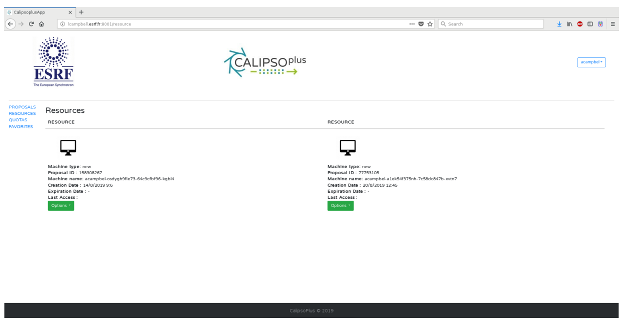

Figure 3: Container and Virtual Machine which the user has made and can use.

Site Specific Login

Each facility will be able to implement additional authentication methods to their local site portal. UmbrellaID is the preferred authenticator for accessing the portal.

Containers

Containers are currently being run on Kubernetes by using the Kubernetes Python API which supports mounting directories the user would need such as their home directory or their Proposal directories.

In order to access user home directories inside the container, we initially encountered the problem whereby the container was run as root while trying to access a directory from NFS which belongs to a different user. To access these directories, we had to change the container so that it is always being run by the user and never as root. This served the purpose of not only as a possible way to access the data, but also as a security mechanism as even if a user was able to escape the container, they would only have user rights on the host machine instead of root rights.

Jupyter Notebooks

Initially Jupyter Notebooks were provided by using Kubernetes which creates an individual container for each user. The user is able to access all of their directories as with any container. They were also given them sudo access within the container/notebook enabling them to install any additional software they need. This seems to be the most efficient solution as we are able to scale resources depending on users and keep the users in an isolated environment. One unseen problem was that by running the container as the user, the user's account information didn't exist within the notebook (/etc/passwd). Although this initially appears as a cosmetic issue, software that requires a username during the installation, e.g., to create or configure files etc fail. One possible solution is to add the user's account information to the /etc/passwd directory during start-up.

Another means of providing the notebooks was to use SLURM. SLURM allows the user to reserve resources in a computing cluster in order to run a Jupyter Notebook. This solution is a very useful alternative to Kubernetes however it does have some limitations. Firstly, users are not able to install additional software which requires root privileges. The other being that it is initially more difficult to configure as each node of the cluster which can run the notebook must have all of the needed software installed which is not needed for Kubernetes as the container will contain everything.

Data Catalogue

A very important issue is that the user is able to access all of the data relating to their proposal. The default granularity supported will be at the level of Proposal. More detailed data searches for datasets will be done via the data portal and not via the common DAAS portal. It is up to the facility data portal to make a link to the DAAS portal services. The critical issue of making data accessible in an efficient way via a local mount. The type of mount will depend on the local facility file system(s). At the ESRF it was possible to mount the Proposal directory to the container/VM giving the user the quickest possible access to their data. Other facilities are currently doing the same.

TESTING

The development of the portal included a testing phase which reported (in month 24 of the project) critical feedback for assisting further advancements of the system. The examination was done against the Blueprint requirements but also reported the views of developers, system administrators and end-users. The main challenge was the very high number of interconnected software components and services, many of which are still under development, that consist of the platform. Even if the project has common objectives, each facility may have slightly different needs from the portal and some have more advanced deployments than others.

The structure of the examination included functional and non-functional aspects. The functional part included Unit, Integration, System, and Acceptance testing; and surveyed many components such as the login, front-end, analysis applications and data access. This included examining the readiness for delivery and wide deployment. The second part included the so called non-functional aspects that include Performance, Security, Usability, and Compatibility testing. This part provided insights on the overall responsiveness but also raised security concerns.

The results of the testing confirmed that the system as a whole follows successfully the blueprint that it is based on.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA057

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

The facilities with advanced installations which involved end-users during the testing phase, reported a mostly positive and encouraging feedback. A key critical finding is that even if the basic installation is straight forward, the integration to the existing computing infrastructure, services and databases is very challenging due to the complexities of the local infrastructures. Certain issues were raised regarding documentation that since that time have been improved including authoring of the present paper that aims at serving as a reference to the whole system. It was noted that the connection to the data catalogue was a task that required further work; likewise for the local authentication subsystem. There was positive feedback regarding the overall performance but system administrator groups raised the need for specialised monitoring tools. This will pave the way for extending the portal into a production-ready services for end-users as it is planned from certain facilities.

The overall positive feedback of the testing suggested that the system is suitable to serve as a starting-point and demonstrator for EOSC related projects like PaNOSC [14] and EXPaNDS [15].

USER FEEDBACK

Development versions of the portal were deployed in all of the facilities to get user feedback. The users selected to test the portal were scientists and scientific computing group leaders as these groups were considered potential users of the portal in the future.

The overall feedback from users about the tool was very positive. *"As it is now the portal provides all that is required to run a virtual machine..."*; *"I've found the portal very attractive and responsive, and very useful to our users"*. Nevertheless, there were some concerns regarding the performance as, in some cases, the application tested was *"a bit slow, and I had difficulty in browsing the data"*, and this should be addressed in order to adjust every container to its optimal configuration (number of CPUs, RAM,...).

Other comments were received related to new features, such as being able to install any application easily, and being able to use a console or terminal in order to interact with the system.

FUTURE DEVELOPMENTS

The CALIPSOplus DAAS prototype has been useful to identify and solve technical issues, prototype use cases, and get feedback from users. The question now is what is the next step? One obvious goal would be to run the prototype with the aim of developing a production solution in the long term. For this to happen each site needs to adopt a data analysis strategy and to provide human and financial resources to implement this. Most CALIPSOplus partners are however not this far yet - even if all of them admit there is a data analysis bottleneck, few of them have adopted a strategy nor allocated sufficient resources to deal with the problem. One partner of CALIPSOplus is planning to move the DAAS prototype to a production ready service while the other partners will continue to test the DAAS portal as an incubator service.

Fortunately the European Commission has embarked on the European Open Science Cloud (EOSC) [8]. The EOSC is an ambitious project to make scientific data FAIR and to link data to compute resources. Although the exact implementation EOSC is still not clearly defined a number of projects have been financed to further define and implement the EOSC. Two of these projects are the Photon and Neutron Open Science Cloud (PaNOSC [14]) and the ExPaNDS project [15]. PaNOSC is a consortium of Photon and Neutron sources on the EU roadmap, while ExPaNDS is made up of a consortium of national EU photon and neutron sources. Both projects have the goal of providing data services. The future development of the CALIPSOplus JRA2 portal will be carried out under these two projects. See paper [TUBPL02] on PaNOSC at this conference for more details about the data services and portal developments that are planned.

CONCLUSION

From this blueprint, we can see that the proposed architecture and implementation of providing Data Analysis as a Service for synchrotron facilities provides a common user experience and services while still allowing the freedom for each synchrotron to implement site-specific functionality depending on their own hardware and experimental procedures. Initial prototypes and tests show that the main issues required to provide a DAAS service for synchrotrons have been taken into account. There are multiple factors like data access, performance, compute resources which need to be taken into account but which can only be done at the facility infrastructure level. Ultimately the service needs to be tested by the users. This will be done through the user meetings and organisations. Each facility must now define the level of support they want to give for Data Analysis as a Service. This will determine the hardware requirements for the production system.

ACKNOWLEDGEMENTS

The authors acknowledge the support and feedback from all of the CALIPSOplus JRA2 collaborators - A.Roux, J.Kieffer, T.Vincent, A.Solé, A. de Maria, M.Retegan, V.Favre-Nicolas, B.Rousselle (ESRF), A.Ashton, T.Schoonjans (DLS), M.Ounsy, G.Viguier (SOLEIL), J. Reppin, F.Schluenzen (DESY), B.Schramm, M.Grobosch (HZDR).

REFERENCES

[1] "Calipsoplus." `http://www.calipsoplus.eu/`. [Online; accessed 16-September-2019].

[2] "ERA." `https://ec.europa.eu/info/research-and-innovation/strategy/era_en`. [Online; accessed 16-September-2019].

[3] "Cross site use case requirement report including comparison of existing solutions." `http://www.calipsoplus.eu/`

ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-WEPHA057

wp-content/uploads/2018/09/D24-3_reported.pdf.

[4] "PaNdata Software Catalogue." https://software.pan-data.eu/. [Online; accessed 22-August-2019].

[5] F. Perez and B. E. Granger, "Project jupyter: Computational narratives as the engine of collaborative data science," *Retrieved September*, vol. 11, no. 207, p. 108, 2015.

[6] "Umbrella." https://www.umbrellaid.org/. [Online; accessed 26-August-2019].

[7] "OpenID Connect." https://openid.net/connect/. [Online; accessed 26-August-2019].

[8] "EOSC." https://ec.europa.eu/research/openscience/index.cfm?pg=open-science-cloud. [Online; accessed 4-September-2019].

[9] "Kubernetes." https://kubernetes.io/. [Online; accessed 22-August-2019].

[10] T. A. White, R. A. Kirian, A. V. Martin, A. Aquila, K. Nass, A. Barty, and H. N. Chapman, "CrystFEL: A software suite for snapshot serial crystallography," *Journal of applied crystallography*, vol. 45, no. 2, pp. 335–341, 2012.

[11] O. Mandula, M. Elzo Aizarna, J. Eymery, M. Burghammer, and V. Favre-Nicolin, "PyNX. Ptycho: A computing library for x-ray coherent diffraction imaging of nanostructures," *Journal of Applied Crystallography*, vol. 49, no. 5, pp. 1842–1848, 2016.

[12] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PloS one*, vol. 12, no. 5, p. e0177459, 2017.

[13] "Django." https://www.djangoproject.com/. [Online; accessed 26-August-2019].

[14] "Photon and Neutron Open Science Cloud." https://www.panosc.eu. [Online; accessed 10-September-2019].

[15] "ExPaNDS." https://expands.eu/. [Online; accessed 10-September-2019].

© Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.