DOI: 10.17199/NOBUGS2016.65

Building a prototype Data Analysis as a Service : the STFC experience

Frazer Barnsley, Brian Matthews, Tom Griffin, Jody Salt, Derek Ross, Catalin Condurache, Alexander Dibbo, Shirley Crompton
STFC Rutherford Appleton Laboratory, Chilton, Didcot, OX11 0QX, UK

E-mail: frazer.barnsley@stfc.ac.uk

Abstract. In this paper, we describe the motivations behind the design of the Data Analysis as a Service platform, under development within STFC. This is a service aimed at providing a platform for users to access compute and software resources suitable for analyzing experimental data arising from the use of large analytic facilities, for example ISIS and Diamond Light Source. We give some requirements for the use of such a platform, and then go on to discuss its initial architecture. We then describe the current state of the implementation of this architecture with an initial example of its use. We conclude with a discussion on the future direction of this development,

1. Introduction

Modern instruments and detectors are capable of capturing large amounts of data in one scan, and experiments are becoming more sophisticated, with multiple techniques applied concurrently or dynamic structures such as chemical reactions being observed. Data volumes have now grown so large that in many cases it is simply not practical for users to transport the data to their home institution. In other cases, the analysis chains are complex, with a combination of data analysis and simulation, requiring access to high performance computing, large memory machines and a complex software stacks for effective and timely processing of data. These resources and expertise may not be consistently available to all users. As a consequence, many facilities are exploring how to best provide additional computing resources to enable users to access and analyse their data remotely. At STFC Rutherford Appleton Laboratory, there is co-location of large-scale facilities with a dedicated data-centre hosting large-scale data archives, computing capabilities and specialist expertise. This gives the opportunity to coordinate into one service access to in- and post-experiment computing support for facility users to aid those users to interpret their experimental data.

STFC's Scientific Computing Department is working with the RAL based facilities (ISIS, Diamond, CLF) to implement and deploy a *'Data Analysis as a Service'* system (DAaaS) to support this service. Such a system provides facility users with easy to use access to compute resources, collocated with the experimental data archives, to efficiently and easily process their data, within a managed, secure virtual environment. Commonly used software packages will be systematically made available via a deployment and configuration system, and the environment offered to users will be customised according to the nature of the experiment and requirements of the experimental team. The system will further support a number of interfaces, allowing both easy access to users for routine tasks as well as a more interactive environment for more specialised usage.

In this paper we describe some requirements that such a DAaaS platform should be expected to satisfy. We then go on to describe the architecture of our approach, and our experience of configuring and deploying a prototype DAaaS for facilities at STFC's Rutherford Appleton Laboratory. Finally, we discuss our plans to extend and develop this system to provide a production environment, covering a range of analytic techniques to users within one service

2. Requirements

The high-level requirements the DAaaS platform should satisfy include the following.

1. Users should have access to their experimental data and compute resources for data analysis during their visit and after they return to their home institution.
2. User access should be secure; they should be able to access the data for their experiments and only their experiments. Other users should not be able to monitor a user's activity within their data analysis environment.
3. Users should be provided with a suitable software environment. That is, current analysis, visualisation, workflow and other software appropriate to their scientific approach should be made available to them systematically within the environment, with suitable licencing. Users should be able to upload their own software packages; however, the quality of such software should not impact other users.
4. Users should be able to access the compute resource appropriate for the current task. Thus, if they are in-visit it may be appropriate to access instrument workstations or facilities clusters, while post-visit they should be able to access data-centre compute resources. If the compute task becomes too large for one system, they should be able to submit jobs onto a HPC cluster.
5. As users access different compute resources, the data available to them should be consistent – that is, the current working data should move as the user accesses different platforms transparently to the user (subject to latencies in moving the data).
6. Users should be provided with user interfaces (virtual desktops, specialised clients, web-clients) suitable to their science domain, community knowledge base and level of expertise.
7. The system providers should be able expand the service to additional experimental user groups at marginal cost – that is, it should be a software deployment and configuration task to introduce new groups rather than a complete redevelopment. Thus the system can be efficiently delivered across many communities.

These requirements are demanding; we propose that the most appropriate basis to build a DAaaS system satisfying these requirements is on a loosely-coupled and expandable cloud-based platform.

3. Design of the DAaaS

We discuss the overall design of our prototype DAaaS. A cloud-based solution can provide the basic platform; this would provide an environment which can be remotely accessed, be securely insulated from other users, and be provisioned with access to the relevant data and software. The existing capability within Scientific Computing Department can provide such a platform, which we are using to develop the DAaaS. An outline of the DAaaS architecture is given in Figure 1.

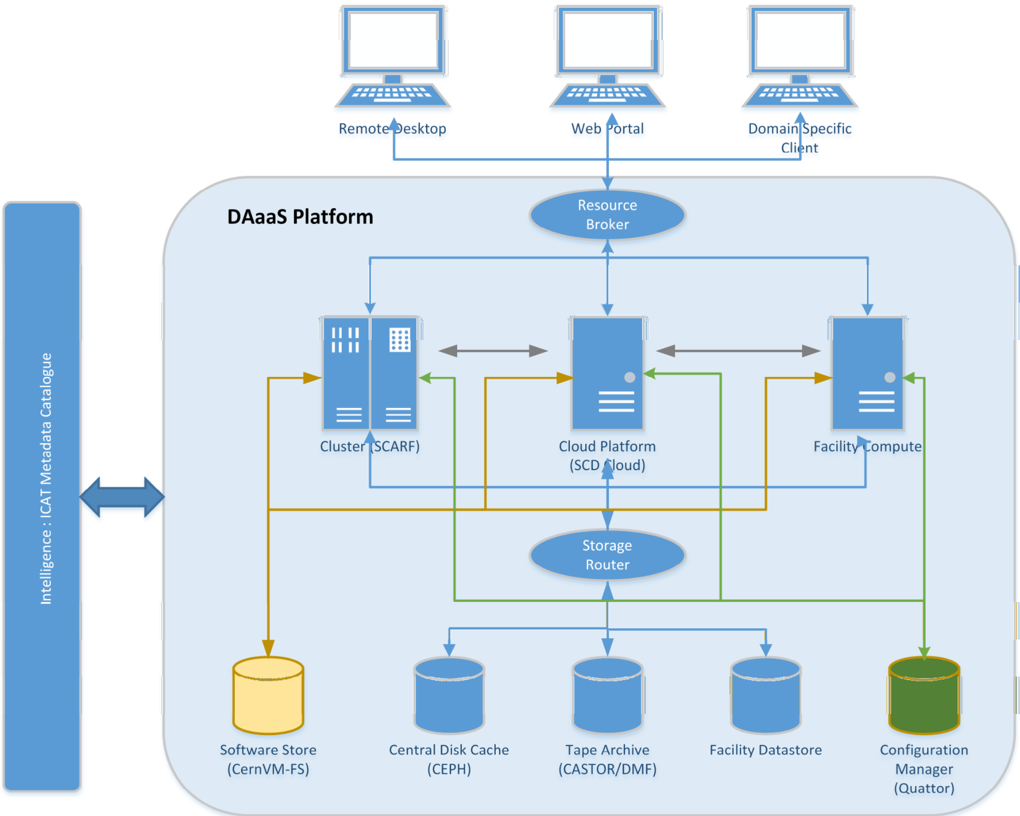

Figure 1 : Outline architecture of Data Analysis as a Service

The architecture can be split into three main layers:

Storage Layer: the "data backplane" with the persistent data storage available to all the tools, services and software within the DAaaS, on disk or tape, located centrally or within facilities (e.g. on instruments). Data should be maintained consistently between them, and made available to the compute services.

Compute Layer: the compute services which are available to the users, with a number of different compute platforms. The user should be able to move between them easily and have their data made available on local caches, consistent with their last use of the data.

User Interface Layer: a number of different user interfaces should be made available to the users, depending on their familiarity with the systems and the nature of the interactions which they would require with the Platform. Thus we can provide a direct access via a remote desktop, giving the user highly flexible and customisable access to the platform. Further, we can provide more controlled environments via web portals or domain specific user clients.

The Platform would also integrate services which customised and configure the platform to the needs of the particular user domain, and also allow the deployment of common software packages, so that users have access to consistent versions of software within the different compute platforms, with appropriate libraries and configuration.

The Platform includes a notion of a controlling source of "intelligence". That is, it uses information on the users, users' experiments, and experimental domains to contextualise the service. This controls access to users to their experimental data, but also guides the system to provide access to appropriate services. For example an experimental group in imaging would be given access to suitable reconstruction software, with licensing appropriate to the user, and use of resources such as high-performance clusters suitable to the scale of reconstruction, and access to workflows and visualisation tools. The system will also maintain the information, collecting provenance data as the users process their data to provide an audit trail for tracking and restarting progress through the analysis, for replaying workflows and for validating results. The source of intelligence within the STFC DAaaS is the ICAT metadata catalogue [1], a catalogue of facilities experiments and associated data.

4. The Technology Stack of the DAaaS prototype

The technology stack we are using to prototype the DAaaS system is guided by our existing experience with Cloud systems, distributed storage systems such as Ceph, software distribution and packaging using CernVM-FS, and our experiences of different methods for providing remote desktop like services. We give an overview of the status of the components used in the DAaaS.

Compute Layer: The platform currently has two main compute resources - a cloud compute cluster, based on OpenNebula [7] and SCARF, a batch system based on IBM's Platform LSF [8]. The cloud has 896 processing cores and 3.5TB of memory and provides users with interactive analysis environments via customised virtual machines (VM). The SCARF batch system has approximately 6000 cores coupled with 300TB high speed storage based on PaNaSaS [9]. It has over 70 different applications available and more than 500 users. Jobs can be submitted through an API provided by Platform LSF. The default remote entry point to the DAaaS system is via the OpenNebula cloud platform.

User Interface Layer: The platform provides a web portal for users to interact with the platform. It allows them to launch interactive analysis environments customised to specific science domains. Users currently have three methods to access interactive environment via remote desktop – NoVNC [2], SSVNC [3] and RDP [4]. NoVNC is a browser based client for VNC connections. It makes use of the HTML5 canvas element and WebSockets. It requires no setup or configuration for the user and can provide a full-screen remote desktop session in browser. RDP is a remote desktop technology, available on Windows systems by default and there are clients for Linux and OSX as well. It provides better performance than NoVNC for visually intensive work or connections across large geographic areas but it requires more steps from the user to set it up. The final method is using SSVNC. This is a secure VNC client that works across Windows, Linux and OSX. Again, it provides better performance than NoVNC but requires that the user installs a client.

Storage Layer: The platform provides three different method for users to access their data - through the browser, WebDAV [10] and Globus / GridFTP [11]. The browser based data access allows users to upload and download data into the analysis environments. WebDAV is supported out-of-the-box on Windows, Linux and OSX and allows users to map a network drive. For large scale data transfers, we provide a Globus Online endpoint. This technology is based on GridFTP that is able to handle the transfer of very large data volumes in a stable way.

Software: Software is made available via a mixture of CernVM-FS and a configuration management system, Quattor. CernVM-FS is a network file system based on HTTP that is designed to deliver scientific software onto virtual machines and physical worker nodes [5]. Some of its features include aggressive caching, digitally signed repositories and a user defined hierarchy for the software. It is capable of supplying multiple versions of software over different operating systems and architectures and has the ability for automatic file de-duplication. CernVM-FS is useful for scientific applications that have been designed in a portable way where dependencies are packaged into the software. When software is placed in a CernVM-FS repository it is then usable by both the VM analysis environments and the SCARF batch system. For software that is not portable, we use Quattor [6]. This is a system administration toolkit providing a portable and modular set of tools for the automated installation, configuration, and management of clusters, farms, grids and clouds. This is used to configure and install the software. Quattor is also used to configure the different types of analysis environments.

5. An example: tools for Macro-Crystallography.

As an initial example of the use of the DAaaS platform, SCD has been working with the Collaborative Computational Project on Macromolecular Crystallography (CCP4) to host tools developed by CCP4 in the prototype cloud platform. This is described in detail in a separate presentation at NoBugs 2016 [12]. A screen shot of the DAaaS remote desktop with the CCP4 tools is given in Figure 2.

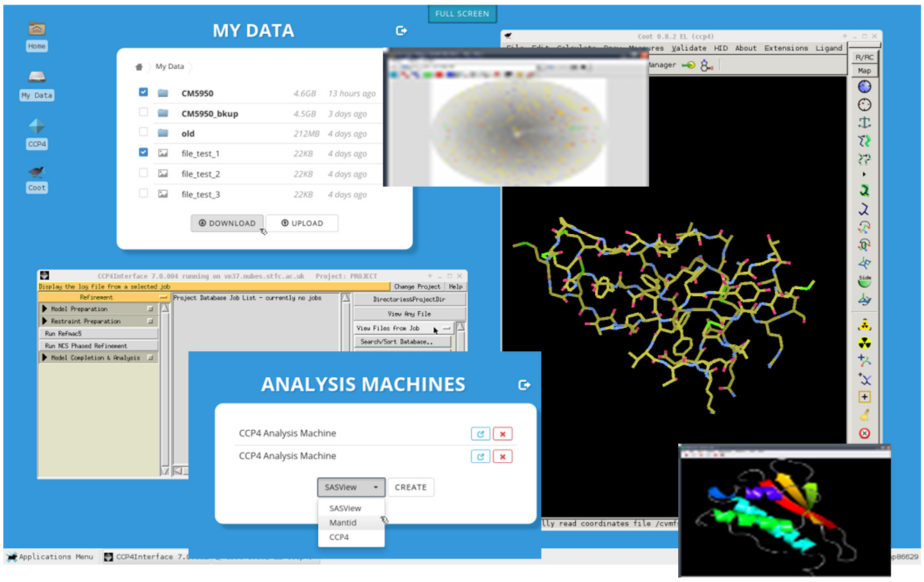

Figure 2 : the DAaaS remote desktop with CCP4 tools.

Registered CCP4-DaaS users can access CCP4 virtual machines via a remote desktop or a browser client. The CCP4 VMs are configured at the contextualisation stage which mounts the user's persistent file share and provides a set of standard software tools including the CCP4 integrated suite of programs

6. Future plans

The current prototype DAaaS platform provides a useful demonstrator and a testbed to validate and fine tune our initial requirements. Work continues to develop some of the more complex functionality of the system, as well as expand the range of software environments and user interfaces the system offers to different experimental user communities.

A particular challenge is the storage router. The router needs to co-ordinate the flow of the users' experimental data from the (disk and/or tape) archives to the cache of the compute service that the user is currently using. Then current data needs to be moved between compute platforms as the user changes their focus. To do this seamlessly and invisibly to the user requires a careful analysis of how data is used and what is required to keep the state of the individual data caches consistent with the local state of the data. Data then needs to be stored in persistent storage and then the caches flushed.

The use of the CernVM-FS to maintain consistent versions and images available to all platforms also can be integrated into a continuous integration software engineering process. The CernVM-FS can be supplied by continuous integration platforms such as Jenkins. In this case, as new software versions are released, they can be rapidly deployed onto the DAaaS platform.

The intelligence core of the system, embodied within the ICAT needs developing and expanding. In particular the integration of the ICAT with customisation services to provide tailored user environment needs to be developed. Further use of tracking and provenance to monitor and guide the usage of the system within particular analysis workflows needs to be explored.

Acknowledgements

We would like to thank our colleagues in SCD particularly in the ICAT, WLCG Tier 1 and CCP4 teams, and in ISIS and DLS for their help and advice in the development of the DAaaS. The work is supported by ISIS and SCD within STFC. We would also like to thank the wider PaNData community for providing an arena to share and discuss ideas in this emerging area.

References

[1] Stephen M Fisher et. al. The growth of the ICAT family. *NoBugs 2016, Copenhagen, Denmark, 17-19 October, 2016.*

[2] VNC client using HTML5 (WebSockets, Canvas) with encryption (wss://) support. https://kanaka.github.io/noVNC/

[3] SSL/SSH VNC viewer http://www.karlrunge.com/x11vnc/ssvnc.html

[4] An open source remote desktop protocol(rdp) server. http://www.xrdp.org/

[5] CernVM File System (CernVM-FS) https://cernvm.cern.ch/portal/filesystem

[6] Quattor http://www.quattor.org/

[7] OpenNebula - Flexible Enterprise Cloud Made Simple http://opennebula.org/

[8] IBM Spectrum LSF http://www-03.ibm.com/systems/spectrum-computing/products/lsf/

[9] Panasas - High Performance Parallel Storage http://www.panasas.com/

[10] Web-based Distributed Authoring and Versioning http://www.webdav.org/

[11] Research data management simplified | globus https://www.globus.org/#

[12] Shirley Crompton et. al. Virtual Large Facility for Experiment-based Structural Biologists. *NoBugs 2016, Copenhagen, Denmark, 17-19 October, 2016.*