EPiC Series in Computing

Volume 95, 2023, Pages 249–256

Proceedings of European University Information Systems Congress 2023

Bringing HPC clusters into Science Mesh

Holger Angenent[1], Daniel Müller[2] and Raimund Vogl[3]

[1] University of Münster holger.angenent@uni-muenster.de
[2] University of Münster daniel.mueller@uni-muenster.de
[3] University of Münster rvogl@uni-muenster.de

Abstract

Data transfers to and from High Performance Computing (HPC) clusters are a rather well established mechanism since standard Linux file transfer tools like *scp* and *rsync* can be used. Nowadays, users are not always familiar with these command-line based mechanisms and prefer more user friendly, and web based solutions. In this article we present a system to use the protocols of *enterprise file sync and share systems* (EFSS) for data transfers utilizing components of Science Mesh. In addition, this offers a technique to transfer data between different HPC sites without the need for commercial software or exposing the rather vulnerable port 22 to the outside world.

1 Introduction

The University of Münster operates the HPC cluster *PALMA II*, a one petaflops machine with almost three petabytes of data. On the one hand, the goal is to make it possible for users to migrate their data easily to other HPC sites within the state of North-Rhine Westfalia in the scope of the HPC.NRW project[1]. On the other hand, to enable data transfers to the EFSS sciebo[2] shares used by all North-Rhine Westfalian universities. Due to network restrictions, other HPC clusters cannot be directly reached via the usual transfer protocols like *scp* and *rsync* using port 22, since this also enables users to gain interactive access via the command-line. Nowadays, access via ssh is therefore mostly restricted from outside the institution for security reasons.

[1] https://hpc.dh.nrw/
[2] https://www.sciebo.de

2 Data transfer in HPC clusters

When it comes to transferring data, most HPC clusters behave like a traditional Linux computer offering a command-line interface to a standard unix shell. Data is normally transferred via port 22, which means users can invoke file transfering tools like *scp* and *rsync* if using any unix-based operating system (Linuxes, BSDs, MacOS etc.) or GUI applications like *WinSCP* when using a Windows computer.

scp is a simple and secure file transfer protocol, offering high performance due to its very low overhead. *rsync* offers advantages in terms of bandwidth efficiency, since it has a delta transfer feature. Its *resume* capabilities also come in extremely handy. Another common tool, *Rclone*, is more focused on cloud storage and therefore supports many different protocols. It also offers a synchronization feature like rsync does. The higher complexity of Rclone compared to scp and rsync however, might cause problems especially for beginners.

All these tools are rather robust, well tested and performant, but rely on having direct access to port 22 of the target system. For security reasons, connections via port 22 to and from many HPC sites are restricted and not reachable world wide, making it mandatory to use a VPN client, if the researcher's computer is not located within the network of the institution hosting the cluster. For example, this often is the case when researchers are granted computing times at other HPC centers.

However, installing VPN clients might not be allowed or even possible depending on the hosting site. In addition, storing data sets in the range of multiple terabytes on a researchers local workstation is often rather slow or not even possible due to the sheer amount of capacity needed. Another restriction using the methods described, is the necessity of an account on both computers. Transferring data from researcher A on computer 1 to researcher B on computer 2 is most often not possible.

One tool to overcome at least some of these restrictions is the commercial, and in parts proprietary, solution *Globus Connect*. If it is available on both clusters, data can be transferred very performant - even without direct access to port 22.

In this article, we introduce a method that enables data transfers from and to HPC clusters via a bundle of tools that include *Nextcloud* and *Rclone*. Within Science Mesh, the integration of Rclone is one of the major tasks.

3 What is Science Mesh?

Science Mesh[3] is a federated cloud mesh connecting heterogeneous sites, their different types of Enterprise File Sync & Share (*EFSS*) systems, and ultimately their users in a transparent way.

Besides seamless sharing of data between different file sync and share systems, Science Mesh offers data science environments like *JupyterLab*, federated use of applications for collaborative document editing like *Collabora* and *OnlyOffice*, as well as fast and reliable on-demand high volume data transfers.

[3] https://sciencemesh.io, https://cs3mesh4eosc.eu

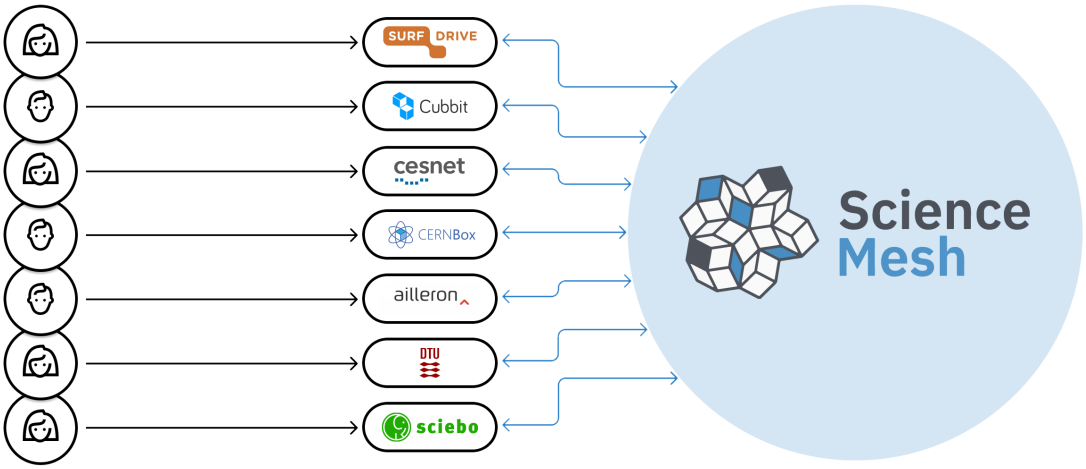

Figure 1: General Science Mesh Structure

3.1 Vision

Science Mesh aims at connecting users of its constituent sites, allowing them to share their work and collaborate on it - even if their home systems are of entirely different type.

Before Science Mesh was brought to life, jointly working on a project was usually possible only by either using the same EFSS system, forcing most users to create and maintain a secondary account on an external system, or to manually share the files and data by other, even more cumbersome means. This made collaboration unnecessarily difficult and people had to waste valuable time on tasks not directly related to the actual project.

Science Mesh now enables all fellow workers in a project to always stay within their home system, allowing them to both independently and cooperatively work on the project in an effortless fashion. Through offering a multitude of technologies specifically targeted towards scientific research, like data science environments, high speed data transfers and collaborative document editors, the goal of the Science Mesh project is to boost open science and to help researchers truly focus on what really matters - science.

3.2 Use cases

Many scientific sites are already part of Science Mesh. Below, an overview of selected use cases showing the various areas utilizing the technologies brought by Science Mesh can be found.

CERN - High Energy Physics The Large Hadron Collider produces unprecedented volumes of data, and the involved data analysis tasks are performed by distributed teams of scientists from all over the world. Because of this, one of the biggest challenges in High Energy Physics is providing streamlined tools for effective collaboration in this distributed environment.

By integrating data science environments such as *JupyterLab*, Science Mesh facilitates collaborative research and enables federated sharing of computational tools, algorithms and resources.

Thus, users are able to access remote execution environments without the need to set up and handle additional accounts on external systems.

RiseSMA - Coordinating papers and managing social media The RiseSMA project develops solutions for social media analytics (*SMA*). Their many tasks include coordinating incoming papers and managing their social media content, which requires seamless synchronous editing of documents across different systems.

The collaborative editors offered by Science Mesh help RiseSMA to perform these tasks without the need to manually share files and relying on emails to send data.

LOFAR - Transferring large amounts of data The Low-Frequency Array (*LOFAR*) is a large radio telescope network consisting of a vast array of omnidirectonal antennas. LOFAR stores its collected data at three storage locations and processes them at four compute locations. It is therefore essential to have an efficient, fast, reliable and user-friendly way to move these data sets from system to system.

By implementing high speed, high volume data transfers, Science Mesh greatly helps LOFAR to move its data sets between their storage and compute locations.

3.3 Becoming part of Science Mesh

First of all, every institution running a supported[4] EFSS system can join Science Mesh for free. There are a couple of formal and technical steps required to join the mesh, which are all outlined in detail at:

https://developer.sciencemesh.io/docs/how-to-join-sciencemesh/

Since the project is still in its development stage and the actual process is subject to change, the necessary steps are not listed here.

3.4 Technical implementation

Science Mesh not only connects EFSS systems that otherwise would be unable to share data between each other, it also offers many additional services that support researchers in their daily collaborative tasks. This was made possible by developing and integrating various interoperability protocols and application programming interfaces (*APIs*). Furthermore, albeit the constituent sites of the mesh are autonomous, a central component is used to manage the mesh metadata and to perform monitoring to ensure a smooth operation of the mesh in its entirety.

OCM and the CS3 APIs Two cornerstones of Science Mesh are the *Open Cloud Mesh* (*OCM*) protocol and the *Cloud Services for Synchronization and Sharing* (*CS3*) APIs.

OCM[5] is a vendor-neutral open protocol which offers a common file access and data transfer layer across different sites, regardless of the underlying EFSS system. It is used to let these systems talk to each other in a well-defined, streamlined way.

[4] Currently, ownCloud and Nextcloud are supported, support for Seafile is in progress.
[5] https://wiki.geant.org/display/OCM/Open+Cloud+Mesh

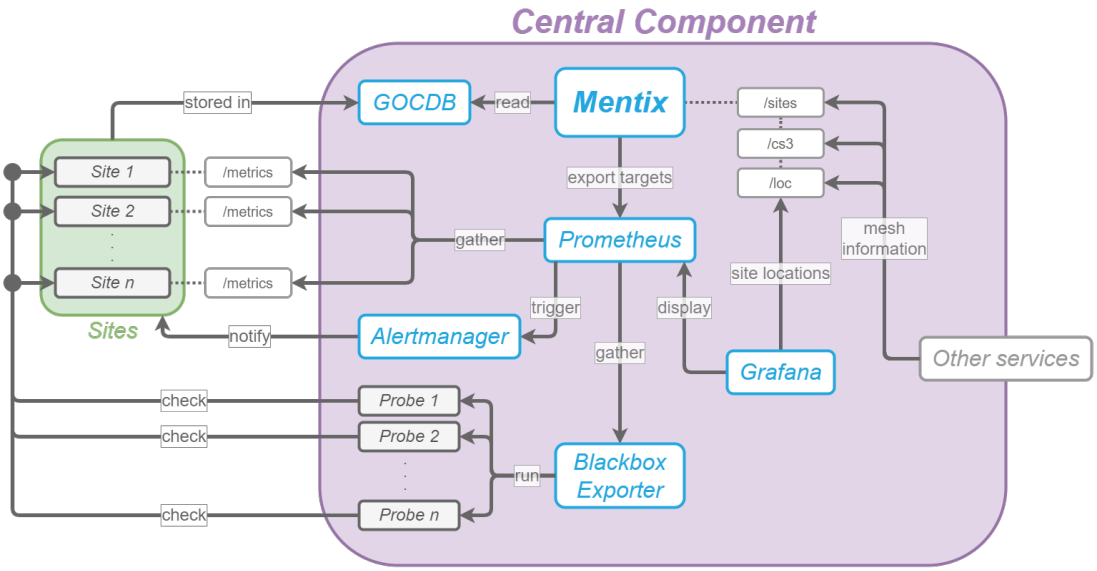

Figure 2: The Central Component

The CS3 APIs[6] on the other hand build a common interface used by both the underlying storage and application providers to communicate with each other. They are used to bind together the nodes and applications within Science Mesh.

Reva The Reva project[7] is a standard implementation of the CS3 APIs used to provide interoperability of different EFSS systems, thus forming a standard vendor-neutral platform that can be readily used by all participants of Science Mesh. It leverages the CS3 APIs in order to offer a straightforward, portable and scalable way to connect existing services.

The Central Component The central component is used to store the mesh metadata, including information about the sites that are part of it and the services they offer. It also performs various monitoring tasks, like gathering statistics and performing health checks, to ensure a smooth operation of the mesh as a whole.

While a complete description of every part of the central component is beyond the scope of this document, figure 2 shows a schematic overview of the various services it contains and how they are related.

The **GOCDB** is used to store all the metadata, which is then consumed and processed by **Mentix**, a service that mainly performs automatic configuration of other services in the central component. **Prometheus** and **Grafana** are used to collect and display metrics, while a customized version of the **Blackbox Exporter** performs periodic health checks on all sites and their services. In case of any warnings or errors, the **alert manager** notifies the administrators of the affected site.

[6] https://doi.org/10.1051/epjconf/202024507041
[7] https://reva.link

While this document is being written, the components are hosted at the University of Münster with a backup site at CESNET.

3.5 Realizing high speed, high volume data transfers

Commercial solutions like Dropbox or Google Drive allow to easily share data with people who are either part or not part of the same organisation. However, they allow data *sharing* but no data *transfer* - this is where Science Mesh comes into play.

In Science Mesh two types of data transfer are available :

- **Adhoc data transfer:** simply the manual transfer of electronic files from person to person via email, instant messaging or, in recent times, through file-sharing apps. Here, data is copied from one storage system to another. This is important for the longtail users, that is, individual people that need to transfer from one place to another.
- **Orchestration of data transfer:** automates processes related to managing data, such as bringing data together from multiple sources, combining it, and preparing it for data analysis. This one is more focused on international communities and big organisations. The tools used in the background are Rucio, Rclone and the File Transfer System (FTS)[8]. In Science Mesh these allow high performant, orchestrated data transfers without the need of using a command line.

4 How to make an HPC cluster a Science Mesh node

Figure 3 shows the concept of how different nodes of Science mesh can exchange data with each other. Here, the node on the right hand side is a traditional EFSS (like many organizations are hosting one) that has an additional Reva daemon running to make it a Science Mesh node. The other two nodes shown, share a major difference. Their back-end storage is not exclusively in use by the EFSS, but is a standard parallel file-system like GPFS, Lustre or BeeGFS as it is used for HPC systems. To integrate it into a Nextcloud, it needs to be exported on one node via SMB. This export must not be exposed externally for security reasons and will only be used via Nextcloud that can include SMB mounts and show them to the users. SMB can be configured such that it exports a different path per user so that each user will see an individual path like **/scratch/tmp/$user/transfer**, where **$user** stands for the username. Special care has to be taken for the user management, as SMB does not recognize local users from an LDAP, but has to be a member of an AD for example.
Using the SMB external storage integration has the advantage that Nextcloud does not need exclusive access to the storage. Data that is written there by other processes will be recognized by Nextcloud without the need of a manual synchronization.
For the integration into Science Mesh, the Reva daemon and the front end integration app for Nextcloud need to be installed as well.

5 Performance Measurements

To benchmark if the proposed solution is viable for transferring large data sets, performance measurements were made.

[8] https://cs3mesh4eosc.eu/technologies/file-transfer-service-fts

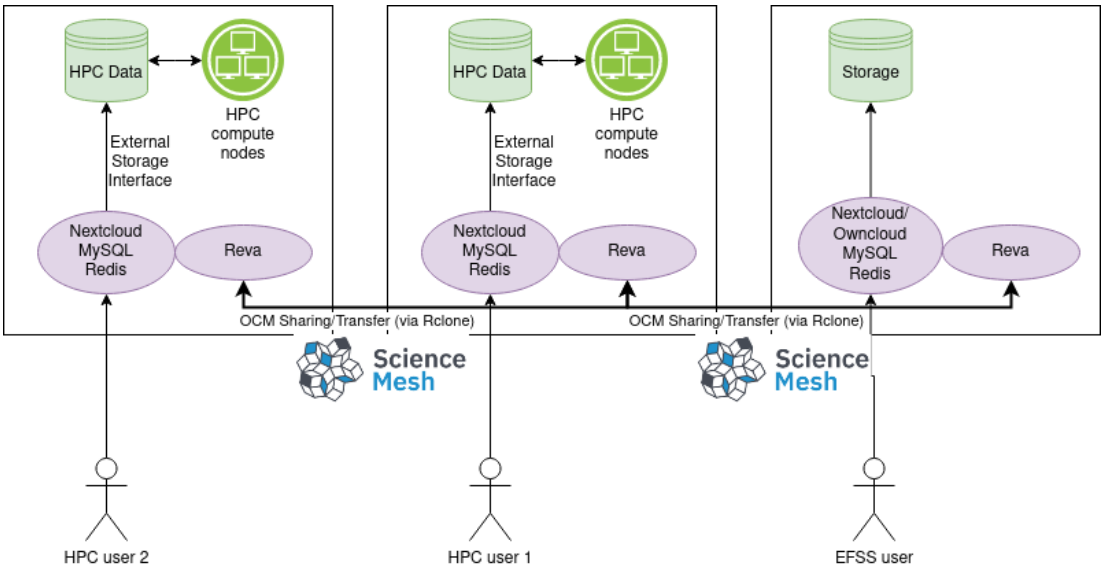

Figure 3: Data transfer between HPC clusters and Science Mesh. Data sharing is established via the OCM protocol in the background. Data is transferred via the Science Mesh implementation of Rclone.

Unfortunately, the data transfer part of Science Mesh is not in production at the time of writing, so no tests between two instances could be performed. Nevertheless we tested the upload speed of data transfers via the web interface. Since there are some additional layers that are not present in simple scp or rsync based transfers, we expected some slightly slower performance in comparison.

For the test a file of 1 GB was uploaded via the Nextcloud web interface. This is then stored by the system directly on the Spectrum Scale (a.k.a. GPFS) file system of the cluster and can immediately be used. The computer that was used to upload the data had a 1 GBit internet connection.

The performance that could be measured in the Nextcloud case was about 80 MB/s. For comparison, transfers via *scp* and *rsync* were made to the headnode of the cluster in the same filesystem. Via *scp*, a transfer rate of 100 MB/s was measured, while *rsync* delivered a transfer speed of 80-90 MB/s.

Using the webinterface did cost some performance, but only moderately. Further tests are necessary with faster network connections to validate, if the upload via the webinterface saturates faster than via *rsync*/*scp*.

6 Conclusion and Outlook

Our proposed method of data transfer aims at offering a more user friendly approach with additional functionality for accessing data pools on HPC clusters that are traditionally accessible only via command-line based protocols. EFSS solutions like Owncloud and Nextcloud have proven to be able to handle petabytes of data. Nevertheless, more practical testing is necessary

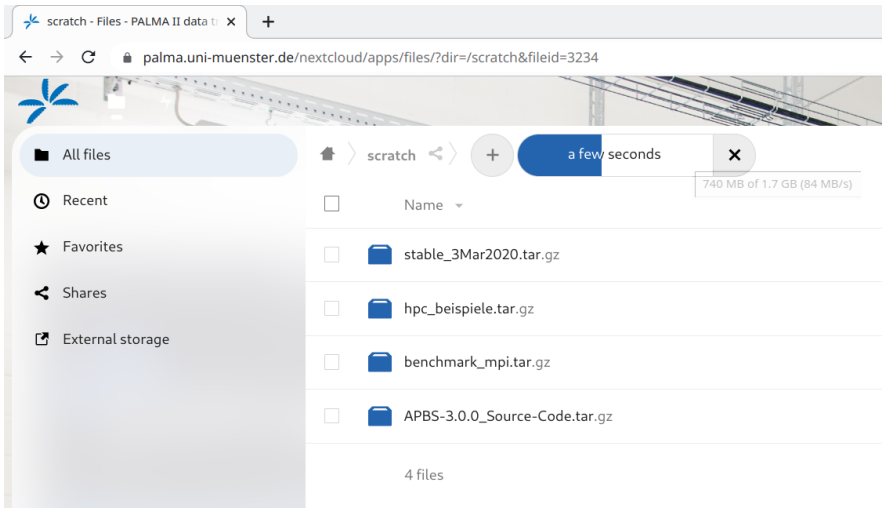

Figure 4: Upload to the scratch directory via Nextcloud

to demonstrate that the method using the external storage interface proposed here scales well enough for HPC data volumes and numbers of files. Also, tests in terms of robustness are to be performed with a larger number of users. If Science Mesh becomes available in production, data transfers to sciebo, the EFSS of the University of Münster, will be the next step into production readiness.