Data Storage Federation for VPH Applications

Koulouzis, S.; Vasunin, D.; Belloum, A.; Bubak, M.

Publication date
2014

Document Version
Final published version

Data Storage Federation for VPH Applications

Spiros Koulouzis[1], Dmitry Vasyunin[1], Adam Belloum[1], and Marian Bubak[1,2]

[1] Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands
contact:{S.Koulouzis, Dvasunin, A.S.Z.Belloum,M.T.Bubak}@uva.nl
[2] Department of Computer Science, AGH Krakow, Poland
contact: bubak@agh.edu.pl

As in many scientific communities, the key challenge within the Virtual Physiological Human (VPH) community is to share and access large datasets that can be used by medical applications to transform them into meaningful diagnostic information. To achieve this, researchers need advanced storage capabilities to enable collaboration without introducing additional complexity to the way data are accessed and shared [3]. The plethora of data access technologies and distribution of data makes their sharing difficult. The fragmentation of datasets makes their discovery complex and time-consuming. Another problem arises from the need to handle a large set of client implementations in order to consume data from different sources. Furthermore, it is unrealistic to force storage providers to install and maintain large software stacks on their data resources in order to provide access to a variety of different clients. For instance in the context of VPH-share domain [2], it cannot be assumed that datasets are located in a single storage infrastructure nor that they are available via a single technology. Moreover, a variety of scientific applications are legacy applications and can only access data from a local disk. Last but not the least the vendor lock-in problem[3] is another issue scientific communities have to consider when using clouds.

A solution that can address these problems is data storage federation. Our approach is to aggregate a pool of resources in a client-centric manner and present it via a standardized protocol that can be also mounted and presented as a local storage so we can provide a file system abstraction. This can be achieved with the introduction of a common management layer that uses loosely coupled and independent storage resources. Through this common management layer a variety of storage resources such as simple file servers as well as storage clouds and data grids[5] can be aggregated exposing all available storage under a unified interface. As a result, distributed applications have a global shared view of the whole available storage space. This makes the development, deployment and debugging of applications considerably simpler. Applications can be developed locally and then deployed on a distributed environment without changing the data access parameters. Legacy applications that can only read/write files on a local disk can easily ported to distributed environments. Moreover, the available storage space is used more efficiently with the copy-on-write strategy[4]. Furthermore, replication of data can be based on efficiency cost measures. These measures can range from environmental costs[5] to performance costs[6]. Offering a global shared view of data storage between applications improves the reliability. Experience has shown that a large percentage of execution failures is attributed to missing files [4]. Having a common data management layer can help minimize this problem. When using cloud storage we also have a reduced risk of vendor lock-in since no large amount of data are located on a single provider.

The main objectives of the research presented in this paper is to elaborate an architecture that can flexibly federate clouds grids and other storage technologies, without any changes

[3] Vendor lock-in is a problem that occurs when a user cannot migrate her data from a cloud storage offering, because the price that needs paid for downloading all the data at once, exceeds the benefit of using a cheaper vendor

[4] When multiple separate logical data use identical copies of the same physical data, it is not necessary to create separate copies of that data, instead they can all be given pointers to the same data resource

[5] e.g. How much energy or how much CO2 is required

[6] e.g. store data on near the computation and/or on high performance storage infrastructure

to the existing infrastructure. Existing approaches are either tightly coupled with the infrastructure and thus demand from data owners to use a specific technology or they are oriented towards a specific environment and therefore implement ad-hoc interfaces making them less interoperable. Our main goal is to provide a large scale virtual file system that can take advantage of all available technologies in order to provide a global shared view of the underling resources. At the same time our approach is easy and intuitive to use as it hides the complexity of federation under a standardized access interface.

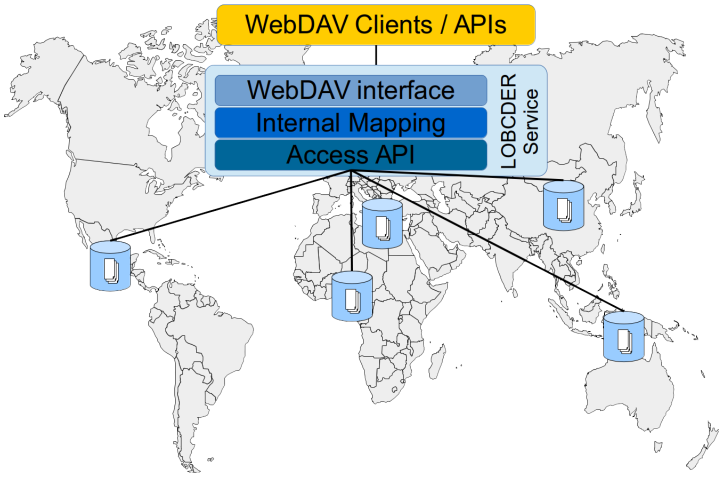

Fig. 1. LOBCDER's architecture. The top layer is a WebDAV repository, the middle layer maps the physical files to a logical file system and the bottom layer provides transparent access to a variety of infrastructures.

The **L**arge **OB**ject **C**loud **D**ata storag**E** fede**R**ation (LOBCDER) is a storage federation service that aims at making available distributed scientific data stored in various storage framework and independent providers. It is a part of the Data and Compute Cloud Platform of the VPH-Share project. Within that framework, LOBCDER needs to expose large amount of data to the VPH community. At the same time, it is important not to add new software on the data storage and provide to users and applications easy and transparent access to distributed datasets. Not adding new software on the data storage is important not only because the complexity of managing and maintaining the existing software stack is already high, but also because some storage is outside our administrative domain and we cannot enforce the installation of specialized software that will allow us to federate that resource. Therefore, LOBCDER loosely couples a variety of storage resource that may use different technologies such as Openstack-Swift[7], iRODS [7], and may be geographically distributed. The LOBCDER is a data management solution that provides a WebDAV interface allowing users and applications to choose a variety of available clients and APIs. Such an approach drastically simplifies access and sharing of datasets that are hosted on different locations and infrastructures. Figure 1 shows LOBCDER's approach to storage federation.

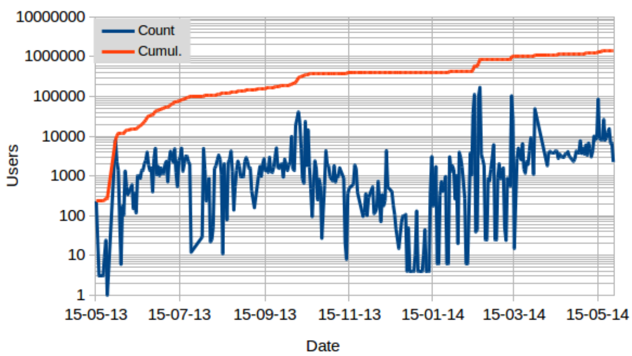

Fig. 2. Users logged in LOBCDER in the past year

In a collaborative and large scale research environment such as VPH-Share, applications need access to large datasets which are geographically distributed and located in different storage infrastructures. LOBCDER, has been operational for almost one year offering a simple and transparent way of accessing and sharing datasets to VPH applications. The @neurIST [6] application is using LOBCDER for storing and retrieving data. Its requirements dictate flexible capabilities for browsing and analyzing data, as well as a consistent set of APIs to access the various data sets it uses. LOBCDER covers these requirements with two interfaces: a WebDAV and a ResFUll interface. The RT3S Vascular Stenting [1] tool is another application

[7] http://swift.openstack.org/

running on the Data and Compute Cloud Platform of the VPH-Share project. This application is using LOBCDER in order to load large images (300 MB) analyze them for peripheral stent fatigue-fracture and produce a fracture risk report which is saved in LOBCDER.

To this date it has granted access to 1.454,999 users[8] and users have performed 11,070 downloads and 71,018 downloads. Figure 2 shows the load of users accessing LOBCDER each month as well as the running total of the number of users over time. Applications and users have transfered 82,088 files in total. Apart form the load LOBCDER has being able to cope with, it has also being used by a variety of clients and APIs. At least 16 different clients and APIs have used LOBCDER in the past year. Figure 3 shows the usage percentage of each client or API. From that figure we can also see 24 % of the clients and APIs is counted as null. This means that apart from standard WebDAV clients (davfs, Mozila etc.) many users have being developing and using their own custom solutions for accessing LOBCDER.

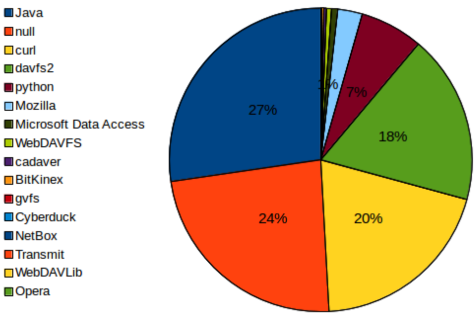

Fig. 3. User agents (clients and APIs) that have used LOBCDER in the course of one year

To address the issues of transparent simple and efficient data access we elaborated a new method of data federation and implemented LOBCDER, a data federating solution able to aggregate multiple storage types and frameworks, and present them to the user as a single storage framework. In order to provide network transparency, a WebDAV repository was developed. The LOBCDER can use cloud and grid infrastructure as well as other technologies as its storage backend. Usage data have shown that LOBCDER is able to cope with large volumes of data and a large amount of requests. Having verified that the current design is able to meet the demands of the VPH community, we will move further in order to implement efficient and flexible replication algorithms as well as caching mechanisms that will improve performance. Additionally, to increase data availability and performance, we will extend our current architecture into a distributed version that will use per-fetching for more efficient cache usage.

Acknowledgement. This work was partly funded by the VPH-Share[9] and COMMIT[10] projects.

References

1. RT3S - Real Time Simulation for Safer vascular Stenting. http://rt3s.eu/.
2. Siegfried Benkner, Jesus Bisbal, Gerhard Engelbrecht, Rod D. Hose, Yuriy Kaniovskyi, Martin Koehler, Carlos Pedrinaci, and Steven Wood. Towards collaborative data management in the vph-share project. In *Proceedings of the 2011 international conference on Parallel Processing*, Euro-Par'11, pages 54–63, Berlin, Heidelberg, 2012. Springer-Verlag.
3. Tony Hey, Stewart Tansley, and Kristin Tolle. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft, 2009.
4. Kyriacos Neokleous, Marios Dikaiakos, Paraskevi Fragopoulou, and Evangelos Markatos. Grid reliability: A study of failures on the egee infrastructure. In Sergei Gorlatch, Marian Bubak, and Thierry Priol, editors, *Proceedings of the CoreGRID Integration Workshop 2006*, pages 165–176, October 2006.
5. Srikumar Venugopal, Rajkumar Buyya, and Kotagiri Ramamohanarao. A taxonomy of data grids for distributed data sharing, management, and processing. *ACM Comput. Surv.*, 38(1), June 2006.
6. M. C. Villa-Uriol, G. Berti, D. R. Hose, A. Marzo, A. Chiarini, J. Penrose, J. Pozo, J. G. Schmidt, P. Singh, R. Lycett, I. Larrabide, and A. F. Frangi. @neurIST complex information processing toolchain for the integrated management of cerebral aneurysms. *Interface Focus*, 1(3):308–319, June 2011.
7. M. Wan, R. Moore, and A. Rajasekar. Integration of cloud storage with data grids. In *Proceedings of ICVCI-09, 3rd International Conference on the Virtual Computing Initiative*, Research Triangle Park, NC, USA, 2009.

[8] Not unique. Each operation in LOBCDER requires authentication
[9] http://www.vph-share.eu/
[10] http://www.commit-nl.nl/