Письма в ЭЧАЯ. 2016. Т. 13, № 5(203). С. 986–992

DYNAMIC FEDERATION OF GRID AND CLOUD STORAGE

F. Furano [1], *O. Keeble* [2], *L. Field* [3]

CERN IT/SDC, European Organization for Nuclear Research, Geneva

The Dynamic Federations project ("Dynafed") enables the deployment of scalable, distributed storage systems composed of independent storage endpoints. While the Uniform Generic Redirector at the heart of the project is protocol-agnostic, we have focused our effort on HTTP-based protocols, including S3 and WebDAV. The system has been deployed on testbeds covering the majority of the ATLAS and LHCb data and supports geography-aware replica selection.

The work exploits the federation potential of HTTP to build systems that offer uniform, scalable, catalogue-less access to the storage and metadata ensemble and the possibility of seamless integration of other compatible resources such as those from cloud providers.

Dynafed can exploit the potential of the S3 delegation scheme, effectively federating on the fly any number of S3 buckets from different providers and applying a uniform authorization to them. This feature has been used to deploy in production the BOINC Data Bridge, which uses the Uniform Generic Redirector with S3 buckets to harmonize the BOINC authorization scheme with the Grid/X509. The Data Bridge has been deployed in production with good results.

We believe that the features of a loosely coupled federation of open-protocol-based storage elements open many possibilities of smoothly evolving the current computing models and supporting new scientific computing projects that rely on massive distribution of data and that would appreciate systems which can be more easily interfaced with commercial providers and can work natively with Web browsers and clients.

PACS: 07.05.Bx; 07.05.Kf

INTRODUCTION

In this work, we report on our activity aimed at providing tools and systems that can fulfill the demanding tasks of grid computing using mainstream protocols that are shared by the Web enterprise world. In this respect Dynafed [1, 2], our dynamic system for managing loosely coupled storage federations, offers very interesting features when used to manage S3 buckets in the context of the grid authentication/authorization. These features, together with the dynamic location of files across site boundaries, have been successfully used to augment the traditional grid data management and storage with resources that can be added and removed opportunistically, while transparently bridging multiple authentication domains.

Among the benefits of loosely coupled storage deployments we can cite an additional resiliency of the whole system towards failure of one of its storage endpoints.

[1] E-mail: Fabrizio.furano@cern.ch
[2] E-mail: oliver.keeble@cern.ch
[3] E-mail: laurence.field@cern.ch

THE DYNAMIC FEDERATION PROJECT (DYNAFED)

The goal of the Dynamic Federation system is to federate local or remote storage sites and metadata endpoints, that expose a suitable data access protocol, into a transparent, high performance storage federation exposing a unique name space. The architecture can accommodate logical file name and algorithmic name translations without the need for catalogues and single points of failure. On the other hand, if catalogues are needed, several of them can be accommodated within the same federation. The idea is to allow applications to access a globally distributed repository, in which sites participate while keeping their autonomy. The applications would be able to efficiently access data that are spread through different sites by means of a redirection mechanism that is supported by the data access protocol used. The focus is on standard protocols for data access, like HTTP and WebDAV, and NFS can be considered as well. The architecture and the components of such a system are anyway decoupled from the actual protocol used.

The system can also accommodate on-the-fly geography-based matching of clients and replicas.

Another important point is that such a system should be efficient also in the browsing case, e.g., allowing the user to list the content of a directory in a fast and reliable way that does not impact the performance of the whole system.

The Dynafed project started in 2011 in the context of the European Middleware Infrastructure (EMI) as an exploration of storage federations with open protocols. Nowadays the core is a stable protocol-agnostic component, which in practice is used only for HTTP, WebDAV, and S3. In this context, it relies only on standard services given by the endpoints and does not require additional components to be deployed there. Being totally standard, among the systems it can interplay well with, we cite the File Transfer Service (FTS) [9, 10], which plays a major role in the LHC computing in order to move in a coordinated way many hundreds of terabytes of data files per day.

FEDERATING GRID AND CLOUD RESOURCES THROUGH HTTP AND WebDAV

Our goal is to integrate seamlessly cloud storage resources in HTTP-enabled workflows subject to the grid authentication schemes, that is, X.509 with VOMS extensions [3]. In other words, we want to use cloud resources together with existing grid and HEP distributed storage and workflows, giving focus to the following aspects:

- no more effort with respect to the normal that is required to administer sites;
- agility in adding or removing local or remote endpoints, with no downtime and catalogue synchronizations needed;
- make usage and management seamless;
- promote scalability, performance and software quality;
- preserve sites' autonomy;
- allow "opportunism" in resource management;
- very simple technical requirements, easy to share with other scientific communities.

HTTP and WebDAV. Although the system is agnostic with respect to the communication protocols used, our focus has been on HTTP-based protocols for the following reasons:

• HTTP has appropriate technical features:

— it is a flexible and extensible protocol which covers most existing use cases, while allowing new stuff;

— applications just access the data, wherever they are (very different from distributed FSs, that are limited to the concept of mountpoint);

— supports WAN direct access;

— performance can be very high for applications using it efficiently;

— it is available in a multitude of software/hardware platforms.

• HTTP is moving much more data than High Energy Physics (HEP) worldwide, although in different ways that in general are less coordinated than HEP;

• HTTP offers the familiarity of browsers, and at the feeling of simplicity they give;

• HTTP for scientific computing is a step towards convergence:

— one technology can accommodate multiple *use cases, also interactive;*

— users can use their preferred *devices* and apps to access their data;

— sophisticated custom *applications* are allowed;

— can be more easily connected to *commercial systems* and apps.

• It is attractive for a professional to be trained in these systems. Moreover, there are greater chances to be understood when it is mentioned.

Interactivity and the Grid Data Management Problem. The reality of a production-grade distributed model is challenging and is often subject to the so-called "*Where is file X problem*". This is related to just locating a resource in a huge index being a challenge for correctness, scalability and lookup speed. Additional challenges include that:

• the index size may be of the order of 10^9 files;

• with tens or hundreds of sites, the normality is that some of them will be unavailable because of some downtime or unscheduled event. These sites should be avoided for reading file X;

• with 10^5 disks the normality is that quite a few are broken in a given moment, and there is some probability that the one hosting *file X* is broken. This disk or site should be avoided for reading file X, if file X is not there;

• cloud storage is not immune to this kind of problems, as it can be added, removed without notice, become unreachable, or even have downtimes.

These challenges add up to the basic complexity of locating the replicas of file X world-wide. We also would like to emphasize that "Where is file X now" is different from "Where is file X supposed to be". Hence, the goal is to reduce the data management cost for finding it.

The Xrootd framework pioneered around 2004 a solution to this problem, based on real-time communication among intelligent agents managing cells of servers [4].

One of the advantages of this approach is that it spreads the lookup load by asking the working endpoints, organized as a B-tree-64. Other strong points of propagating the query to the working endpoints are:

• reaching an endpoint costs just a network round trip. Even through WAN, most of the time it is quicker than a loaded DBMS;

• by construction the responses of the endpoints are correct at that moment, hence the schema naturally models data losses and minimizes the impact of site downtimes:

— at the same time, locations and gathered metadata can be cached for some time, assuming that if a file is accessed now, then it likely will be again accessed shortly (temporal locality principle).

This approach has been successfully demonstrated in very large, distributed production environment [5, 6]. Having a frontend system that is able to locate files by asking the endpoints "Do you have file X in this moment" is one of the main goals of the Dynafed project. Dynafed capitalized on the previous experience to build a system that is dynamic, robust and scalable and that can put emphasis on the ease of integration and user friendliness.

Dynafed is based on an extension of this approach, made more flexible through the introduction of properly designed *location plugins*. In addition to the replica location, Dynafed extends this approach to managing and reconstructing file listings in real time. Together with the broad range of storage and metadata backends used and the user friendliness of HTTP and WebDAV, these features make the system very intuitive to use both for users and system administrators.

MIXING GRID AND S3

Recent improvements of the Dynafed location plugins have been centered on cloud storage. Thanks to the flexibility of the approach, the Dynafed system (described in Fig. 1) augmented with S3 buckets can give unprecedented flexibility to grid data access, fully supporting the grid workflows.

The Simple Storage Service (S3) is a service that was originally introduced by Amazon [7], and among others it offers a REST API that:

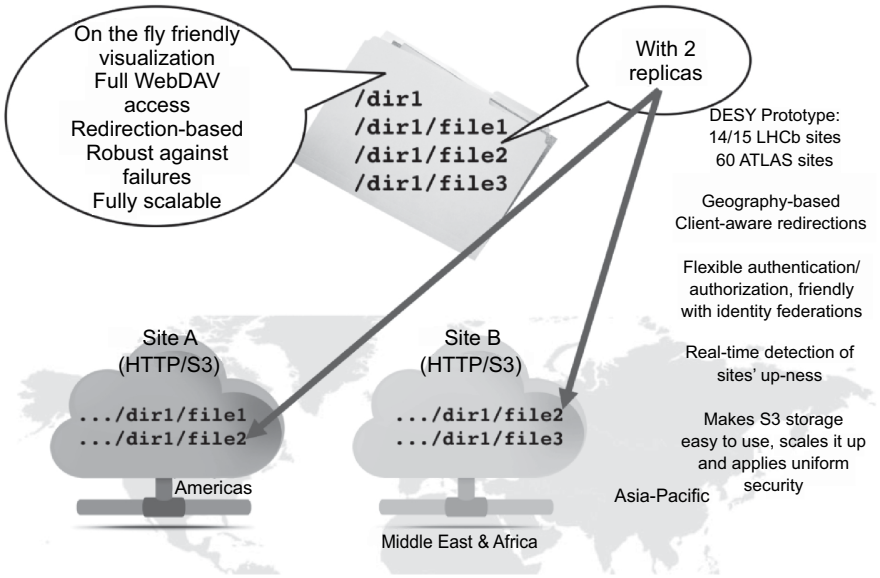

Fig. 1. Dynamic Storage Federation system

• is scalability-oriented on the server side while somehow making nonscalable usage difficult;

• provides a simple and very fast access delegation mechanism;

• supports hierarchical content (similar to directories) in buckets in a way that a vanilla client cannot easily exploit [10].

We wrote a simple *DynaFed* C++ plugin that exploits all these in a friendly way, privileging performance, and flexibility. This gave to the Dynafed system the capability of federating any number of remote S3 buckets together with other non-S3 storages. This mixed federation will work as a unique *read/write* WebDAV storage that is totally seamless, extremely fast and scalable, as it is based on:

• on the fly look-ups to locate files across S3 buckets and grid endpoints;

• short-term caching of the results to enhance the metadata performance;

• redirections to the actual final endpoint, thus avoiding the tunneling of the traffic.

Since this S3 federation is based on signing redirection responses, it will act as a secure authentication gateway, thus avoiding to distribute S3 keys to the clients that need access. The clients will just receive short-term delegations in a redirection response.

Dynafed controls the signing process; hence, it can natively apply a uniform authorization/authentication schema to the whole federation, even if many buckets from different providers compose it. The authentication type can be X509, login/pwd, or in principle whatever mechanism that can work as an Apache module. Hence, users and processing jobs do not need to deal with S3 mechanics or store presigned URLs, as they will just use a clean HTTPS URL and will be authorized transparently by Dynafed throughout the process, as if they were accessing a normal WebDAV very large storage.

Also, the performance is more than adequate, of the order of a few thousand redirections per second per CPU core and is horizontally scalable.

The system has been tested with the S3 implementations of Amazon and CephS3, and a plugin that is able to federate Microsoft Azure is under investigation.

THE DATA BRIDGE

The previous section introduced the interesting features of the Dynafed system when used to federate an arbitrarily large and composite cloud storage.

One would assume that since the authentication of the clients is performed by the Apache frontend, and the Authorization by Dynafed, only one authentication module is loaded.

If, however, the Apache frontend is configured instead with two or more authentication plugins loaded together, the system will act as a storage that accepts multiple authentication protocols and will apply Dynafed's unique, uniform authorization schema before signing a redirection URL to one of its cloud storage backends.

We have named this deployment a "Data Bridge", because clients using different authentication methods can use it to share data in a secure way. The context of the initial idea was High Energy Physics applications running in the BOINC [8] platform for volunteer computing.

The challenge with volunteer computing and the grid was related to the fact that the X509 credentials usually used to access grid resources must not be transmitted to untrusted volunteer computers. In a typical volunteer computing workflow, volunteer users (likely using

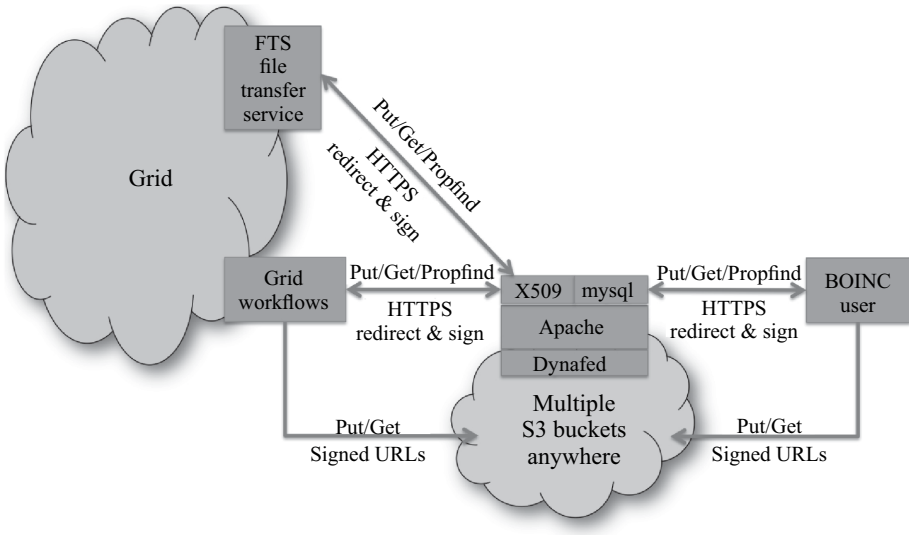

Fig. 2. Federating S3 buckets to bridge authentication domains

username/pwd) need to receive a Job description and be authorized to write the output to a shared area that can potentially become quite large (tens of terabytes); grid agents need to read what the BOINC user wrote to allow validation of the results. This workflow is exemplified in Fig. 2.

CONCLUSION AND FUTURE WORK

Using Dynafed, a system administrator can build a system that transparently uses grid resources and cloud storage. Interface support is limited to S3 in the current version but future work will focus on evaluating Dynafed location plugins that can use other cloud services, for example, Microsoft Azure [2].

The metadata aggregation performance is very high for one frontend machine and is scalable by simply adding parallel machines [1].

We tend to name this approach *HTTP ecosystem*, referring to an ensemble of components that sustain each other's usage and are very open to usage by professionals that do not necessarily have a High Energy Physics background.

The flexibility of the approach also emphasizes the possibilities of sharing services with other communities, developing new services and integrating grid services with mainstream tools, components, and services.

As Dynafed is a generic component, we foresee other applications for it, for example, dynamically clustering remote or local file caches. We also encourage collaborations and new ideas.

REFERENCES

1. *Furano F. et al.* Dynamic Federations: Storage Aggregation Using Open Tools and Protocols // J. Phys.: Conf. Ser. 2012. V. 396. P. 032042. doi:10.1088/1742-6596/396/3/032042.
2. Dynafed Home Page. http://lcgdm.web.cern.ch/dynamic-federations.

3. *Alfieri R. et al.* From Gridmap — File to VOMS: Managing Authorization in a Grid Environment. doi:10.1016/j.future.2004.10.006.
4. *Furano F., Hanushevsky A.* Managing Commitments in a Multi-Agent System Using Passive Bids // IEEE/WIC/ACM Intern. Conf. on Intelligent Agent Technology. doi: 10.1109/IAT.2005.95; Source: IEEE Xplore. http://www.computer.org/csdl/proceedings/iat/2005/2416/00/24160698-abs.html.
5. AAA Project Page. https://twiki.cern.ch/twiki/bin/view/Main/CmsXrootdArchitecture.
6. FAX Project Page. https://twiki.cern.ch/twiki/bin/view/AtlasComputing/AtlasXrootdSystems.
7. Amazon Simple Storage Service (S3). https://en.wikipedia.org/wiki/Amazon_S3. http://aws.amazon.com/s3/.
8. BOINC: Open-Source Software for Volunteer Computing. http://boinc.berkeley.edu/.
9. FTS3: New Data Movement Service for WLCG // J. Phys.: Conf. Ser. 2014. V. 513, No. 3. P. 032081. doi: 10.1088/1742-6596/513/3/032081.
10. *Devresse A., Furano F.* Efficient HTTP Based I/O on Very Large Datasets for High Performance Computing with the Libdavix Library // Big Data Benchmarks, Performance Optimization, and Emerging Hardware. Lecture Notes in Computer Science 8807. P. 194. doi:10.1007/978-3-319-13021-7_15.