Federating Distributed Storage For Clouds In ATLAS

Berghaus F, Casteels K, Di Girolamo A, Driemel C, Ebert M, Furano F, Galindo F, Lassnig M, Leavett-Brown C, Paterson M, Serfon C, Seuster R, Sobie R, Tafirout R, Taylor R P

Frank Berghaus, G07810, CERN, CH-1211 Geneva 23, Switzerland

E-mail: `frank.berghaus@cern.ch`

Abstract. Input data for applications that run in cloud computing centres can be stored at distant repositories, often with multiple copies of the popular data stored at many sites. Locating and retrieving the remote data can be challenging, and we believe that federating the storage can address this problem. A federation would locate the closest copy of the data on the basis of GeoIP information. Currently we are using the dynamic data federation DynaFed [1] software solution developed by CERN IT. DynaFed supports several industry standards for connection protocols like Amazon's S3, Microsofts Azure, as well as WebDAV and HTTP. Protocol dependent authentication is hidden from the user by using their X509 certificate. We have setup an instance of DynaFed and integrated it into the ATLAS Data Distribution Management system. We report on the challenges faced during the installation and integration. We have tested ATLAS analysis jobs submitted by the PanDA production system and we report on our first experiences with its operation.

1. Introduction

Our goal is run data-intensive applications on globally distributed opportunistic resources that have no local grid storage. The ATLAS experiment leverages a globally distributed system of infrastructure as a service clouds, such as Amazon's Web Services, or the Compute Canada, and CERN OpenStack. These resources are integrated into the ATLAS distributed computing system using two cloud scheduler [2] instances: one at the University of Victoria and one at CERN. These IaaS resources are used opportunistically, and do not support any local grid infrastructure.

The workflows executed by high energy physics experiments often demand large volumes of input data or produce a significant volume of output data. We aim to use a data federation, such as DynaFed, to redirect the applications running on opportunistic resources to the optimal storage endpoint to retrieve input or deposit output data.

2. Conceptual Design

The ATLAS experiment leverages the resources of the Worldwide LHC Computing Grid, WLCG [3]. The computer centres that are part of the WLCG and support that ATLAS experiment each host some of the experiment data and simulated events. They provide a global storage infrastructure. While the central ATLAS computing infrastructure uses purpose specific protocols to access the content of these grid storage elements, they may be accessed using standard protocols such as WebDAV, HTTP, and NFS. Figure 1 shows how DynaFed

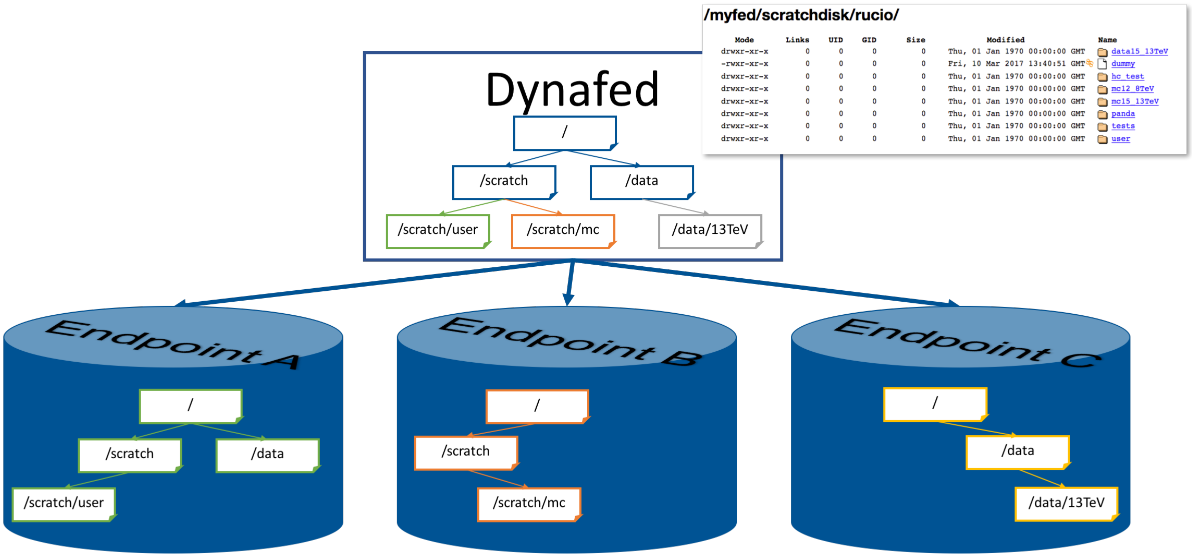

Figure 1. The dynamic federation is cpnnected to multiple endpoints. Each endpoint may be a file system or an object store accessible using the a protocol which allows redirection. The dynamic federation appears to provide a namespace that is a union of all the namespaces of the endpoints. That namespace is presented as a familiar directory structure on the same protocols as exposed by the endpoints.

could appear to present the entire ATLAS data catalog by unifying the namespaces of attached storage elements.

Cloud storage systems are object stores that expose a well defined interface over HTTP and WebDAV, such as Amazon's S3 and OpenStack SWIFT. DynaFed also allows the inclusion of these cloud storage into the WLCG. DynaFed implements grid authentication through X.509 with VOMS [4] extensions and has endpoint plugins that allow it to authenticate over S3, SWIFT, Azure, and so on. Thus a grid user or application authenticates to dynafed using grid crednetials and is forwarded to a pre-signed URL that permits access to the cloud storage system. So, in addition to allowing us to forward applications to their optimal input data on the grid, DynaFed allows the inclusion of object stores which may be local to the opportunistic cloud computing resources.

3. Data Access

4. Application Workflow

5. Summary

Acknowledgments

Authors wishing to acknowledge assistance or encouragement from colleagues, special work by technical staff or financial support from organizations should do so in an unnumbered Acknowledgments section immediately following the last numbered section of the paper. The command `\ack` sets the acknowledgments heading as an unnumbered section.

References

[1] Furano F *et al* 2017 Dynafed `http://cern.ch/lcgdm/dynafed-dynamic-federation-project`
[2] Gable I *et al* 2017 Cloud Scheduler `http://cloudscheduler.org`
[3] Bird I 2011 *Ann. Rev. Nucl. Part. Sci.* **61** 99
[4] Foster I, Kesselman C, Tuecke S 2001 *International Journal of Supercomputer Applications* `http://www.globus.org/alliance/publications/papers/anatomy.pdf`