EPJ Web of Conferences **214**, 04041 (2019) https://doi.org/10.1051/epjconf/201921404041
CHEP 2018

An http data-federation ecosystem with caching functionality using DPM and Dynafed

Davide Michelino[1], *Silvio Pardi*[1,*], Guido Russo[2] and Bernardino Spisso[1]

[1]INFN – Napoli Unit – Via Cintia. 80126, Napoli Italy
[2]GARR – Via dei Tizii, 6 00185, Roma Italy

Abstract. The implementation of cache systems in the computing model of HEP experiments enables to accelerate access to hot data sets by scientists, opening new scenarios of data distribution and enable to exploit the paradigm of storage-less sites. In this work, we present a study for the creation of an http data-federation ecosystem with caching functionality. We created plug-in integrated in the logic of a DPM Storage, able to reproduce a cache behaviour, taking advantage from the new feature introduced in the last version of Disk Pool Manager, called volatile-pool. Then we used Dynafed as lightweight federation system to aggregate a set of standard Grid Storage together with the caching system. With the designed setup, clients asking for a file present on the Data-Grid are automatically redirected to the cache, if the cache is the closest storage, thanks to the action of the geo-plugin run by Dynafed. As proof of the concept, we tested the whole system in a controlled environment within the Belle II computing infrastructure using a set of files located in production Storage Elements. Preliminary results demonstrate the proper functionality of the logic and encourage continuing the work.

1 Introduction

The usage of HTTP and WebDAV as access protocols is becoming a popular investigation topic within High Energy Physics experiments. Today all most used storage technologies provide such kind of support, allowing to exploit a set of new setup, even thanks to the availability of new tools and software created to take advantage of the specific features of HTTP. Among them, Dynafed[1] developed at CERN, represents one of the emerging technologies to easily federate distributed endpoints presenting them as a single namespace. The mount point is accessible via HTTP/WebDAV, offering at the same time a very effective user-experience.

Another helping tool for testing and investigate new storage setup is provided by the DPM (Disk Pool Manager)[2-3] which recently offers a new feature called Volatile Pool. Such feature allows a Storage Element (SE) admin to create special storage area, providing cache-like behaviour driven by custom scripts that can retrieve file from external resources.

Tools just described allow promoting a more dynamic way to use distributed storage resources that goes beyond the simple change of the access protocol. However, in order to

* Corresponding author: spardi@na.infn.it

really integrate those technologies in the computing model of an HEP experiment, we must proof that all mentioned tools could be properly used by the experiment framework, supporting each step of a job workflow.

In this paper, we propose an implementation that combines Dynafed lightweight federator together with the DPM volatile pools for the realization of an HTTP-ecosystem with caching support.

As proof of concept, we created a testing setup in the contest of Belle II experiment [4], implementing all the needed modification in the DIRAC framework to host properly the federator and managing the caching system.

The rest of the paper is organized as follow: in section 2 we summarize the state of the art, in section 3 we present the designed model and the expected feature. In section 4 we show the implementation of the caching system. Section 5 contains some early state tests, section 6 the integration in DIRAC. Finally, in section 7 we summarize our work and discuss the conclusions.

2 Related works

Furano et al. have presented the idea of an HTTP-ecosystem in the paper [5]; more specifically in this work they introduce new features offered by DaviX library which has been presented as first building block to use http-endpoints properly in HEP computing models.

Dynafed, as a new tool for data federation has been presented in [1] and its usage is became quickly very popular in the contest of WLCG communities [6]. In addition we quote the literature related to the investigation for the deployment of new caching systems and their relative integration in applicative workflows, some works has been presented in the context of different experiment like ATLAS and CMS [6][7]. We started our investigation from those works, by designing a model that create an http-ecosystem in federated environment with cache support through an unedited setup which combine Dynafed and DPM Volatile pools. Starting from our testbed, we tried as well to define a model.

3 The designed model and the proposed implementation

Our model can be represented with a four layers stack (figure 1) in which the bottom layer is represented by the set of distributed storage, that can be accessed directly or through the cache layer. Then the federation layer is responsible to create a single namespace and to redirect clients against the most convenient storage for each file request. Finally the application layer include the family of all applicative software that want to access to the data grid.

The realization of this model is possible thanks to the availability of several tools, which implement caching and federation features as summarized in section 2. As proof of the concept, we created a first testbed in the context of Belle II experiment. More specifically we implemented each layer using a set of already existing components combined with a new setup: At application level, we have gbasf2, the command line client for submitting grid-based Belle II jobs, then we used Dynafed as federator, DPM Volatile Pool as caching system and Belle II production storage at lowest level.

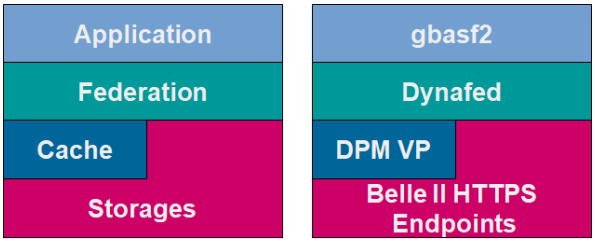

Fig. 1 On the left we show the model stack, while on the right we present the testbed implementation

4 The Belle II Testbed

The testbed implements all the layers described in the figure 2 using technologies already known in the context of Belle II experiment. One of the core component is DPM/DMLite installation which implements the cache layer taking advantage of the Volatile Pool features. The DPM is running over three bare metal servers: one acting as head node e two as disk nodes hosting the data pool for the permanent storage and the data pool for the volatile storage which provides cache-like features for total storage area of 160TB. In order to get all the latest features DPM version > 1.10.0 has been used. All the servers are connected to a 10Gb/s LAN and have access to a 20Gb/s WAN.

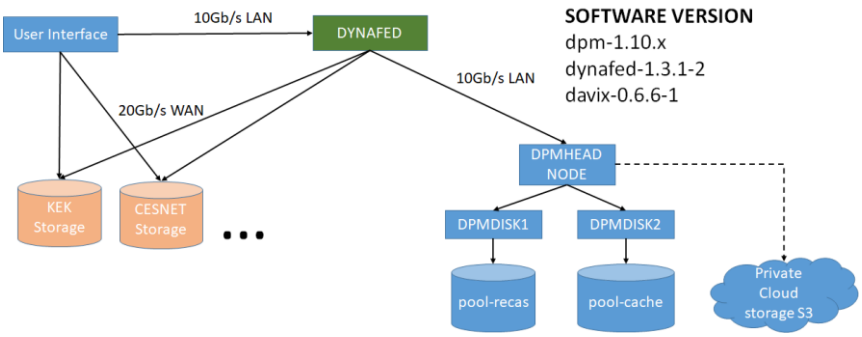

Fig. 2 - Overall testbed architecture.

As federation layer, we used the Belle II testing instance of the Dynafed server running in Napoli since 2016. It currently integrates 19 production storage that cover about the 75% of Belle II endpoints with all the main technologies represented: dCache, DPM and STORM. The table 1 shows the list of all endpoints aggregated in our testbed.

#	STORGE NAME	HOSTNAME	TYPE
1	DESY-DE	dcache-belle-webdav.desy.de	DCACHE
2	GRIDKA-SE	f01-075-140-e.gridka.de	DCACHE
3	NTU-SE	bgrid3.phys.ntu.edu.tw	DCACHE
4	SIGNET-SE	dcache.ijs.si	DCACHE
5	UVic-SE	charon01.westgrid.ca	DCACHE
6	BNL-SE	dcbldoor01.sdcc.bnl.gov	DCACHE
7	Adelaide-SE	coepp-dpm-01.ersa.edu.au	DPM
8	CESNET-SE	dpm1.egee.cesnet.cz	DPM
9	CYFRONNET-SE	dpm.cyf-kr.edu.pl	DPM
10	Frascati-SE	atlasse.lnf.infn.it	DPM
11	HEPHY-SE	hephyse.oeaw.ac.at	DPM
12	Melbourne-SE	b2se.mel.coepp.org.au	DPM
13	Napoli-SE	belle-dpm-01.na.infn.it	DPM
14	ULAKBIM-SE	torik1.ulakbim.gov.tr	DPM
15	IPHC-SE	sbgse1.in2p3.fr	DPM
16	CNAF-SE	ds-202-11-01.cr.cnaf.infn.it	STORM
17	ROMA3-SE	storm-01.roma3.infn.it	STORM
18	KEK-SE	Kek-se03.cc.kek.jp	STORM
19	McGill-SE	gridftp02.clumeq.mcgill.ca	STORM

Table 1 – List of belle II storage aggregated by the Dynafed server in Napoli

The cache logic has been implemented with the help of some scripts, which are triggered by the volatile pools (Fig. 3). Each request to the volatile pool, either a stat or a get request, triggers the script on the head node that resolves the URL of one of the remote storage that has the dataset using Dynafed itself, then checks if the requested path is a file or a directory and reply to the client consequently. The module retrieves as well the size of the file to output proper metadata.

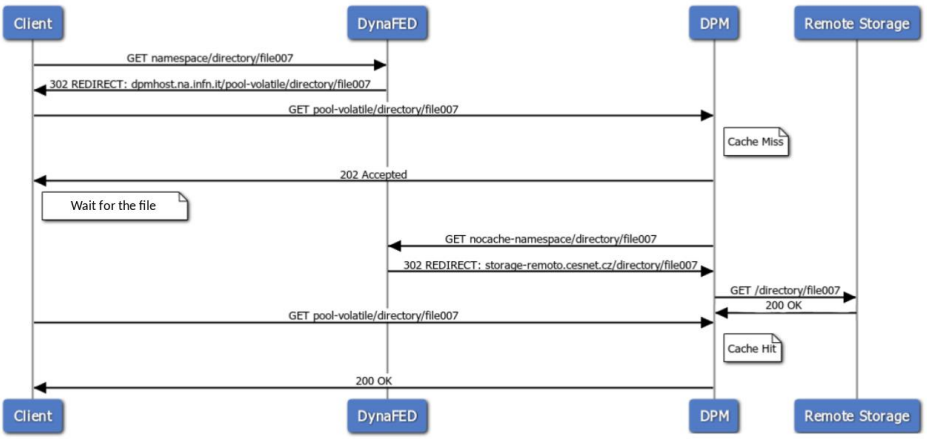

Fig. 3 - Sequence diagram of the cache logic with the federator. A http client requests a dataset to the federator, which routes it to the Volatile Pool. This request triggers the file retrieval from the grid, so the client can download the dataset needed.

The disk node acts only in consequence of a get request. More specifically when a dataset that is not yet in the cache is requested, a helper script is triggered. With the same logic as the script in the head node, the disk node resolves the URL of a remote replica using Dynafed and starts the download from the data grid. While the transfer is in process, the volatile pool returns the 202 HTTP status code (accepted) to the client, asking it to wait for the file to be ready. Common file transfer clients and libraries, used in grid

environment, manage the 202 status code. For example davix or gfal when get such status code, will retry the download after n-seconds (retry delay) up to max_retry attempts.
Since the volatile pool is acting properly as a cache, as soon as it is populated, clients can download directly from it.

5 Testing the main functionalities

As preliminary test, we downloaded from an User Interface in Naples a set of Belle II files, stored in three different continents: Europe, Asia and America. Were used respectively the sites CESNET, KEK and UVic that host Belle II storage (figure 4). Each file set is downloaded three times as follow using gfal-copy command:

- File Download using the direct link to the remote storage
- File Download using Dynafed with Cold cache
- File Download using Dynafed with Warm cache

Tests have been performed using files of 1GB size.

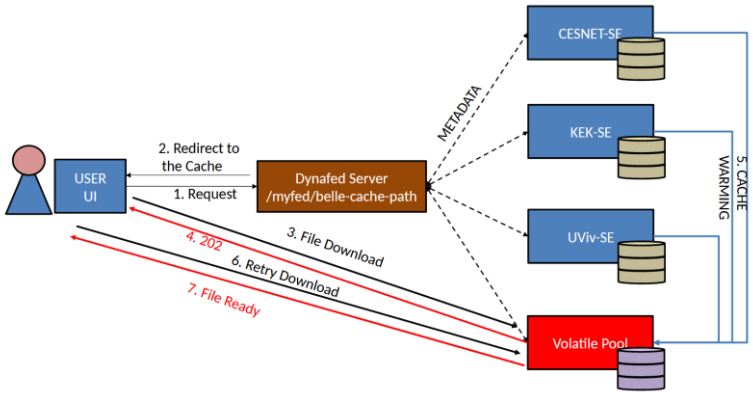

Fig. 4 - Schema of the test. The user located in Naples download a set of remote files accessing via Dynafed. Thanks to the geo-Plugin, the client is redirect against the volatile pool, which retrieves the file from the source before serving it.

Test results show that the testbed is acting as expected from a storage cache, with similar transfer throughput between direct download and from the empty cache and a substantial speed-up when the cache is already loaded with the requested dataset (figure 5).
To be noted that the throughput is calculated with gross times, which include waiting and initialization times and protocols overheads, indeed the aim is to evaluate the actual system performance and not the single transfer throughput.

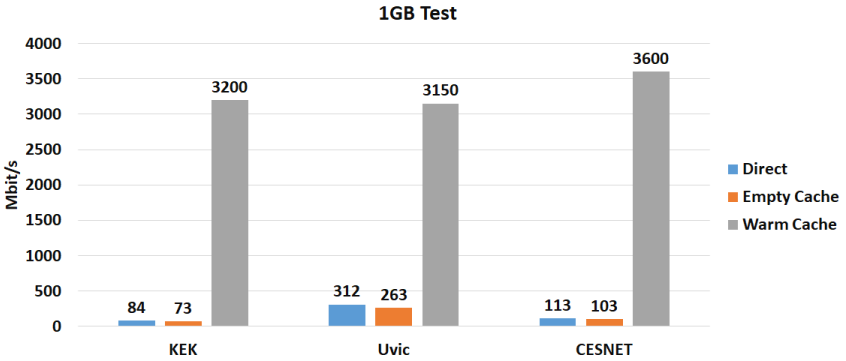

Fig. 5 - Download speed in Mbit/s in the three cases: Direct Access, Empty Cache and Worm Cache.

6 Integration in Belle II DIRAC

In order to properly use the implemented model with gbasf2, we had to integrate it in the Belle II framework for job management, which is based on DIRAC. It required a non-standard registration for storage. For each http endpoint involved in our testbed, we decoupled read and write configuration by using the storage hostname in case of writing and Dynafed hostname in case of file access.
In that way gbasf2 jobs, that access to storage via DIRAC, will access to the federation layer for reading while they will use the direct access to storage for writing. This specific setup was needed in order to be compliant with all the components currently used in the Belle II data model, including file catalogue, metadata catalogue client and file registration procedure.

Using the DIRAC Validation server of Belle II provided by PNNL, we was able to submit jobs via gbasf2, taking advantage from the full http-ecosystem stack, federation and cache included.
As representative test we replicated some datasets in a storage in KEK registered in DIRAC as KEK-DAVS-SE and then we ran a set of analysis job on cloud resources located in Europe using Helix Nebula Science Cloud infrastructure (figure 6). In each job the input files has been read via Dynafed, using the volatile pool feature as well, experiencing the caching effect.

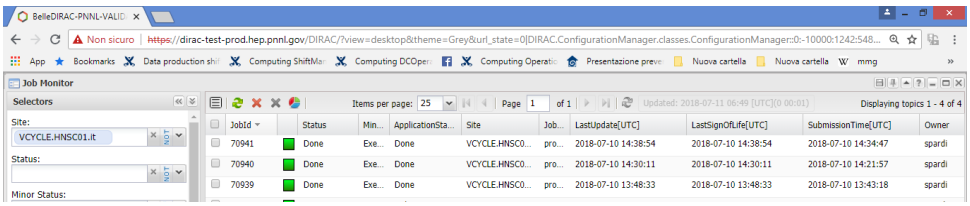

Fig. 6 - Picture shows some jobs done over the Belle II Validation Server using the http-ecosystem

7 Conclusions and future works

In this work, we created a model for a federate http-ecosystem with cache support, meeting the challenges of configuring it with existing tools and integrating it into an existing experiment framework.
The creation of a testbed has been possible thanks to an early adoption of DPM Volatile Pool in combination with Dynafed with a setup never adopted before. The whole system allowed us to proof the feasibility of our approach as test results has shown and at the same time has offered occasion to develop new ideas for further investigation.

Moreover, the integration of the designed http-ecosystem in the Belle II DIRAC Validation Server has represented a first step to evaluate the impact in introducing such innovation in an already established experiment framework. Even if additional works are needed to achieve a full understanding of all possible implication, tests done offer more than a hint about the possible direction to be followed in order to use the designed platform with reasonable and sustainable efforts.

References

1. F. Furano, R. Brito da Rocha, A. Devresse, O. Keeble, A. Alvarez Ayllon and P. Fuhrmann, *"The Dynamic Federations: Federate Storage on the fly using HTTP/WebDAV and DMLite"*, ISGC (2013)
2. F. Furano - *"DPM 1.9 - From the standard flavour to the DOME setup flavour - Features, system architecture and configuration"* - at DPM Workshop (2016) https://indico.cern.ch/event/559673/contributions/2282905/
3. F. Furano et al – *"DOME A rest-inspired engine for DPM"* http://lcgdm.web.cern.ch/dome-documentation
4. S. Pardi et al. *"Computing at Belle II"* - Nucl.Part.Phys.Proc. **273-275,** 950-956 (2016)
5. Fabrizio Furano et al (2014) J. Phys.: Conf. Ser.**513** 032034
6. Pre-GDB of 12 September 12 - https://indico.cern.ch/event/578974/
7. R W Gardner et al (2017) J. Phys.: Conf. Ser.**898** 062017
8. J Balcas et al (2017) J. Phys.: Conf. Ser.**898** 062042

Acknowledgments
Authors wish to acknowledge the assistance of the staff of the ReCaS Data centre, the computing group of Belle II and the site admin of the Belle II production infrastructure.