EPJ Web of Conferences **245**, 04041 (2020) https://doi.org/10.1051/epjconf/202024504041
CHEP 2019

Creating a content delivery network for general science on the internet backbone using XCaches

Edgar Fajardo[1,*], *Derek* Weitzel[2,**], *Mats* Rynge[4,***], *Marian* Zvada[2,****], *John* Hicks[5,†], *Mat* Selmeci[6,‡], *Brian* Lin[6,§], *Pascal* Paschos[3,¶], *Brian* Bockelman[8,‖], *Andrew* Hanushevsky[5,**], and *Frank* Würthwein[1,††] *Igor* Sfiligoi[1,‡‡]

[1]University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093
[2]University of Nebraska, 1400 R Street, Lincoln, NE 68588
[3]University of Chicago, 5801 S Ellis Ave, Chicago, IL 60637
[4]University of Southern California, Information Sciences Institute, 4676 Admiralty Way, Suite 1001, Marina del Rey, CA 90292
[5]SLAC National Accelerator Laboratory, 2575 Sand Hill Rd, Menlo Park, CA 94025
[6]Internet2, 100 Phoenix Drive, Suite 111 Ann Arbor, MI 48108
[7]UW Madison, 1210 W. Dayton St, Madison, WI 53706
[8]Mortgridge Institute, 330 N Orchard St, Madison, WI 53715

Abstract. A general problem faced by opportunistic users computing on the grid is that delivering cycles is simpler than delivering data to those cycles. In this project XRootD caches are placed on the internet backbone to create a content delivery network. Scientific workflows in the domains of high energy physics, gravitational waves, and others profit from this delivery network to increases CPU efficiency while decreasing network bandwidth use.

1 Introduction

Scientific organizations face the problem of delivering data to computing workflows at a combination of owned and opportunistic clusters (sites). Since either the data is read locally at the organization owned sites (no use of opportunistic resources) or the data is accessed remotely with the drawback of high bandwidth utilization and performance degradation of the organization's storage systems (origins).

A third possible solution is the use of caches. Data accessed through the caches travels the network once, from the storage system to the caches. Once cached it can be delivered to

*e-mail: emfajard@ucsd.edu
**e-mail: dweitzel@unl.edu
***e-mail: rynge@isi.edu
****e-mail: marian.zvada@cern.ch
†e-mail: jhicks@internet2.edu
‡e-mail: matyas@cs.wisc.edu
§e-mail: blin@cs.wisc.edu
¶e-mail: paschos@uchicago.edu
‖e-mail: BBockelman@morgridge.org
**e-mail: abh@slac.stanford.edu
††e-mail: fkw@ucsd.edu
‡‡e-mail: isfiligoi@sdsc.edu

multiple jobs thus using the network bandwidth efficiently while acting as a protection layer to the storage systems. This caching strategy implies solving which caching technology to use (Section 2), how jobs access the caches (Section 3), where to place them (Section 4), how to deploy them (Section 4.1) and analyzing their performance (Section 5).

2 Stashcache Background

The Stashcache[1] service is a file block caching technology based on XRootD's [2] caching service XCache[3]. XRootD's architecture is a tree-based structure of servers (bottom leafs) and redirectors (top leafs). Once a client requests a file from the redirector, it queries the servers below it. If they have the file the client is redirected to the server. If not the redirector contacts the redirector above it and so on. This type of tree structure is called a federation.

Stashcache sits in front of a federation. When a client requests a file to the cache, the cache will serve if it is already on disk (cache hit). If not, the cache will contact the federation and retrieve the file from it and then serve it to the client (cache miss). Stashcache has interfaces for the XRootD (propietary) and HTTP(s) protocols and its architecture be seen in Figure 1.

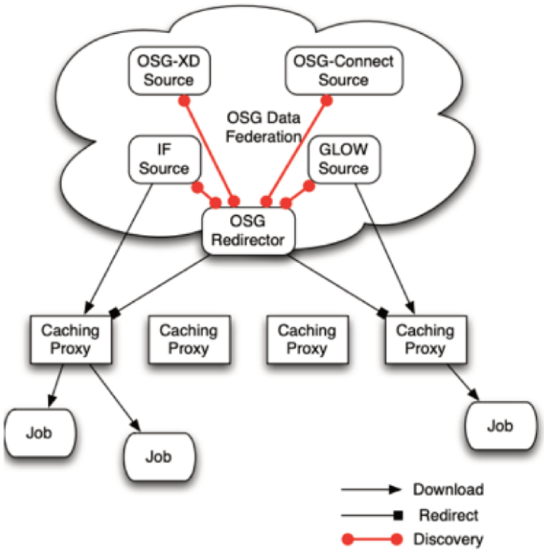

Figure 1. Stashcache architecture.

3 Cache client access via CVMFS

The Stashcache architecture does not mention how users pick which cache to use. The CERN Virtual Machine File System (CVMFS)[4] system solves this problem using GeoIP. It also presents the users a POSIX-like file mount. When a client tries to open a file, CVMFS determines what is the closest geographical Stashcache and retrieves the file from it using HTTP. If the closest cache is unavailable or slowly responding then CVMFS tries the second closest and so on until it retrieves the file. Although CVMFS had been commonly used to distribute scientific software and small files (less than 100 MB experiment calibration data),

the approach to distribute "experiment data" (files over 1 GB) was originally introduced in Ref [5].

3.1 Difference between Stashcache and Frontier squid

Frontier Squid [6] is a widely deployed HTTP-based file caching technology used by CMVFS. There are two major differences between frontier squid and stashcache. A file in a squid cache has a limited lifetime: if the file gets changed in the stratum server (the squid equivalent of an origin), after a few minutes the squid cache will pick up the change and serve the new version. If a file in a stashcache is changed on the origin, that change will not be picked up by stashcache and the cache will serve the old version of the file (until the file is purged from the disk). In other words, the squid model is "write few, read many," whereas the stashcache model is "write once, read many."

The second difference is in the size of files that can be served. Squid caches are optimized for handling small files O(MB) while Stashcache is meant for large files O(GB). Moreover, CVMFS is configured to have compute node level caching of files. This feature is not used by Stashcache however it is used by frontier squid. Files served by Stashcache are of the same size of the local partition hence one file filling the local partition can render useless the compute node level caching.

4 Stashcache Deployment

The OSG through its Virtual Organization (VO) provides several million opportunistic CPU hours for its users and small scientific communities. The computing resources and the data origins are geographically distributed (Figures 2 and 3). Placing caches in the internet backbone reduces the chance that data travels the origin-cache link more than once. Figure 4 show the deployment map of Internet2 and Stashcaches locations in the backbone and at certain academic institutions in the United States. However the International Gravitational Wave Network (IGWN) and the Deep Underground Neutrino Experiment (DUNE) collaborations have their data origins in the US but allocated computing resources abroad hence the need for a worldwide Stashcache infrastructure (Figure 5).

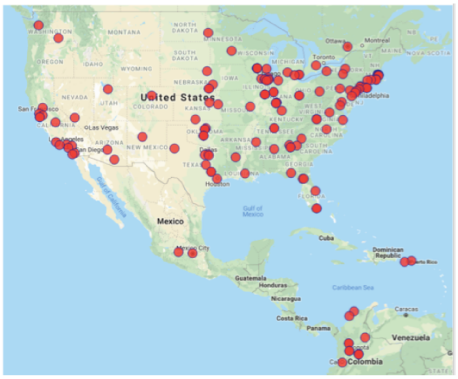

Figure 2. Geographical location of OSG computing resources.

4.1 Kubernetes deployment

Stashcaches are deployed in heterogeneous hardware owned by different institutions. In order to maintain a quality of service and fast turn around from new features to production a

Figure 3. Geographical location of data origins.

Figure 4. Geographical location of Stashcache in the US internet backbone.

Development Ops (DevOps) model is adopted. This approach allows testing of XRootD features and the ability to roll back to previous versions without the intervention of the system administrators.

In this model Stashcaches are deployed as pods using Kubernetes. A pod consists of an instantiated version of a Docker container. The Stashcache containers are maintained by OSG and are routinely optimized and tested to work with new XRootD versions. Kubernetes features volume claims and secrets. The first feature bridges the gap between the stateless micro services that Kubernetes is based on and the stateful XRootD caches; the cache state is the locally stored files. The second feature provides an interface to securely expose X509 certificates to the Stashcache service.

5 Stashcache Performance and Future Work

Stashcache and CVMFS are combined to create a worldwide content delivery network for science. Table 1 summarizes the results of data delivered through the Stashcache network. The second column shows the size of the working set for each organization. The working set is defined as the sum of all the unique files accessed in a time frame. The third column

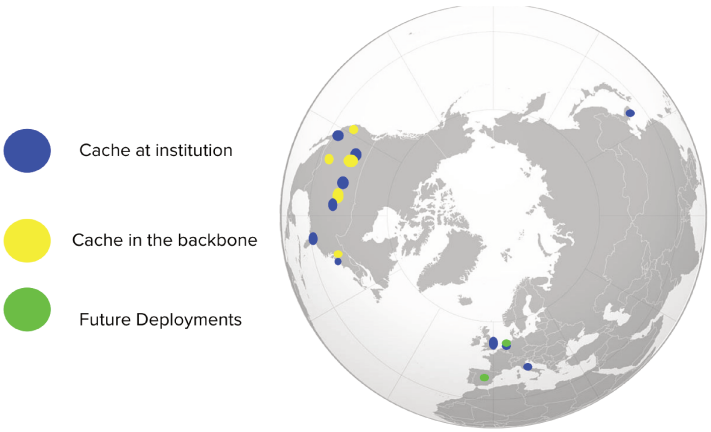

Figure 5. Geographical location of the Stashcache infrastructure

Table 1. Namespace usage from (September 2019 to February 2020) of the Stashcache infrastructure, data obtained from GRACC [8].

Namespace	Working Set (TB)	Data Read(TB)	Data Reuse Factor
DUNE	0.014	1184	84571
LIGO Public Data	7.157	96	75
Nova	0.086	20	233
IGWN	18.172	596	33

shows the total data read by each organization. The fourth one is the division of the third by the second column and represents the number of times the entire working set was read in a time frame. The data reuse factor is an approximation of the cache hit ratio. Future iterations of this work include a file based caching monitoring which provides a more accurate measurement of the cache hit ratio and integration with Scitokens [7] for authenticated access.

Acknowledgement

The authors would like to thank the different funding agencies for this work, in particular the National Science Foundation through the following grants: OAC-1541349, OAC-1826967, OAC-1841530, MPS-1148698. We would also like to thank Internet2 for providing the hardware and support for placing the caches in the backbone.

References

[1] D. Weitzel, M. Zvada, I. Vukotic, R. Gardner, B. Bockelman, M. Rynge, E. Hernandez, B. Lin, M. Selmeci, *StashCache: A Distributed Caching Federation for the Open Science Grid* (2019), pp. 1–7, ISBN 978-1-4503-7227-5

[2] A. Dorigo, P. Elmer, F. Furano, A. Hanushevsky, WSEAS Transactions on Computers **1** (2005)

[3] L. Bauerdick, K. Bloom, B. Bockelman, D. Bradley, S. Dasu, J. Dost, I. Sfiligoi, A. Tadel, M. Tadel, F. Wuerthwein et al., *XRootD, disk-based, caching proxy for optimization of data access, data placement and data replication* (2014), Vol. 513

[4] J. Blomer, C. Aguado-Sánchez, P. Buncic, A. Harutyunyan, Journal of Physics: Conference Series **331**, 042003 (2011)

[5] D. Weitzel, B. Bockelman, D.A. Brown, P. Couvares, F. Würthwein, E.F. Hernandez, *Data Access for LIGO on the OSG*, in *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact* (Association for Computing Machinery, New York, NY, USA, 2017), PEARC17, ISBN 9781450352727

[6] D. Dykstra, Journal of Physics: Conference Series **331**, 042008 (2011)

[7] A. Withers, Z. Miller, B. Bockelman, D. Weitzel, D. Brown, J. Patton, J. Gaynor, J. Basney, T. Tannenbaum, Y. Gao, *SciTokens: Demonstrating Capability-Based Access to Remote Scientific Data using HTCondor* (2019), pp. 1–4, ISBN 978-1-4503-7227-5

[8] K. Retzke, D. Weitzel, S. Bhat, T. Levshina, B. Bockelman, B. Jayatilaka, C. Sehgal, R. Quick, F. Wuerthwein, Journal of Physics: Conference Series **898**, 092044 (2017)