Globally Distributed Software Defined Storage (the proposal)

A. Y. Shevel[1,2,a]**, S. E. Khoruzhnikov**[1]**, V.A. Grudinin**[1]**, O. L. Sadov**[1]**,**
A. B. Kairkanov[1]

[1] ITMO University, Russian Federation, St. Petersburg, Kronverksky prospekt 49.
[2] National Research Center “Kurchatov Institute”, Petersburg Nuclear Physics Institute, Russian Federation, 188300, Gatchina, Orlova Roscha 1.

E-mail: [a] shevel_a_y@niuitmo.ru

Abstract. The volume of the coming data in HEP is growing. The volume of the data to be held for a long time is growing as well. Actually large volume of data – big data – is distributed around the planet. In other words now there is a situation where it is necessary to integrate storage resources from many data centers located far from each other. That means that the methods, the approaches how to organize and manage the globally distributed data storage are required. The distributed storage has several examples for personal needs like owncloud.org, pydio.com, seafile.com, sparkleshare.org. For enterprise-level there is a number of SWIFT distributed storage systems (part of Openstack), CEPH and the like which are mostly object storage. When several data center’s resources are integrated, the organization of data links becomes very important issue especially if several parallel data links between data centers are used. The situation in data centers and in data links may vary each hour. All that means each part of distributed data storage has to be able to rearrange usage of data links and storage servers in each data center. In addition, for each customer of distributed storage different requirements could appear. The above topics are planned to be discussed in data storage proposal.

1. Introduction

In tuning of real storage system there are a lot of aspects and questions even the storage servers are located in one room. The number of nuances is much more in geographically distributed storage systems. The part of them are quite obvious, for example, reliable and secure data transfer over Internet between data storage and client.

In addition there are data transfers to do data synchronization to guarantee the replicas of the data are completely equal each other in different sites. Storage of the data in remote site and data transfer over long distance do require additional precautions such as a data encryption. Data compression and data deduplication are also important to decrease the total data volume to transfer and store.

More and more often we are dealing with big data.

2. Sources of the big data

Among sources of Big Data we have to mention scientific experimental installations:

- ~1+ PB/year - International Thermonuclear Experimental Reactor (ITER - http://www.iter.org);
- ~10+ PB/year - Large Synoptic Survey Telescope (LSST - http://www.lsst.org);

- ~30+ PB/year — CERN – (http://www.cern.ch);
- ~30+ PB/year — Facility for Antiproton and Ion research (FAIR - http://www.fair-center.eu);
- ~30+ PB/year — The Cherenkov Telescope Array (CTA - http://www.cta-observatory.org);
- ~300-1500 PB/year - Square Kilometre Array (SKA - https://www.skatelescope.org)

According to consensus estimation total volume of data in the World grows two times a year, i.e. about 75% of data was written in last two years.

3. Current developments for distributed large volume data storage

Quite a lot of different studies of data storages were presented in [1-3]. Also pretty long list of distributed file/storage systems is available in [4]. A number of developments to meet concrete needs are described in [5-11]. There is a range of studies with presenting benchmarks in [12,13]. The distributed storage is not possible to discuss without network architecture. Software Defined Network (SDN) approach is discussed in [14]. All those sources give the idea which are the main features to be included into the design of a scalable multi-tenant software defined storage (SDS) globally distributed system.

4. Main features of distributed storage

SDS is an evolving concept in which the management and provisioning of data storage is decoupled from the physical storage hardware. SDS should include: Automation – Simplified management that reduces the cost of maintaining the storage Infrastructure; Standard Interfaces – APIs for the management, provisioning and maintenance of storage devices and services; Virtualized Data Path – Block, File and Object interfaces that support applications written to these interfaces; Scalability – Seamless ability to scale the storage infrastructure without disruption to availability or performance.

Any multi-tenant applications running on the public cloud could benefit from the concepts introduced by SDS by managing the allocation of tenant data. Important features of the Globally Distributed Software Defined Storage (GDSDS): *Data store and Data transfer* between client and data storage; also between components of the distributed storage; *Reliability:* data replication, erasure coding; *Reduce the data volume* to store and transfer: *Data compression, data deduplication* (at number of replicas, at level of files, at block level) [15-17]; *Security:* Data encryption, ACL; *User interfaces*: web portal and set of Command Line Interfaces; *Advanced network architecture*; *Optimal Caching, Tiering*; *Automatic storage deployment* by user request.

It is assumed in the proposal:

- GDSDS consists of several groups of storage servers in Data Centers (DC) located in geographically different regions. DCs are connected by a number of parallel virtual data links.
- Data links may have different features: speed, price, encryption type (including quantum encryption [18]), etc.
- Data links have to be configured with SDN.
- GDSDS has web portal for clients and administrators to control and monitoring.
- Client can ask to perform a number of operations:
  - Create, Delete, Replicate, Migrate, etc of the Virtual Storage Instance (VSI) allocated in GDSDS.
  - The VSI can be created with different service level agreement (SLA).
  - Write/Read data to/from the VSI

SLA may include, for example:

- Data Encryption (with specific type of Encryption).
- Data inline Compression (with specific type of compression).
- The use of burst buffers [19-21] between VSI and user applications.
- Number of replicas.
- Erasure coding (specific type or/and parameters).
- In one specific Data Center (DC) or in many DCs with specific types of data links in between DCs.
- Newly created VSI may have character of object storage, file system, or block storage.

- Type of backend (CEPH, SWIFT, etc).
- The preference of access to the geographically distributed data taking into account the context of CAP theorem [22], choose two of three options: consistency, availability, network partitioning.

5. General consideration

The general scheme of the GDSDS is shown in the figure 1. The clients can be also distributed around the World (client can be a person with desktop or smart phone, virtual machine in client data center, also client can be an organization with its own DC, etc). Several clients can use same VSI. In the figure 1 it is shown that all four clients have virtual storage which is located in two or more DCs. Obviously the connection between client and DCs with data located is very important for data transfer speed.

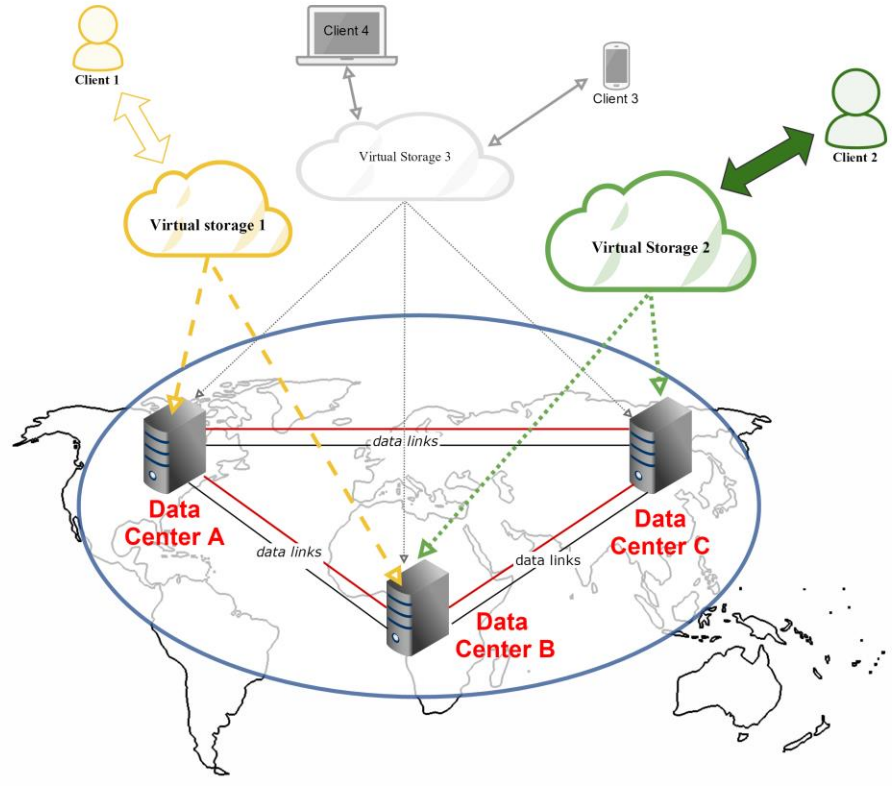

Figure 1. General scheme of the GDSDS.

If VSI has one or more replicas it make sense to allocate data replicas in different DCs. At this point we have to take several aspects into account. First of all we (and client) must plan the procedure of the replicas synchronization. The synchronization of the data between DCs takes time. Do we permit the clients to access the replica during synchronization time or we need to put client request in wait state until the synchronization completes or clients have to be routed to most new replica data? The answer depends on client's business. In some cases client does not care, for example, in Google different clients in geographically different regions obtain different results on the same search request. In other cases clients could have another expectations, for example, they could expect exactly the same result (or very close in some context) on the request from any geographical region.

The synchronization is important when the data in VSI is changing quite often. However even in case the owner did not change the data in VSI the synchronization procedure has to be repeated over some time just to be sure that no errors appeared in data due to any cause. In general, it is necessary to recalculate all hashes in each DC where the replicas are and probably to transfer some fractions of the data to accomplish synchronization. If volume of VSI is around tens (or more) of PB the synchronization time may be significant. What could we suggest clients at this significant time?

Another question how client chooses replica for reading from (or writing to) VSI. It is possible in round robin manner, or just where DC responses faster on the client request. The consideration does emphasize again that there are a lot of various choices for clients therefore we need configurable environment to meet client expectations.

6. Proposal status and plan

A simple prototype was created with only backend – CEPH (version 10.2.5). The prototype has been deployed using several virtual machines (VM). Each VM has 2 virtual CPU, 8 GB of memory, three HDD drives (4 TB each). One drive in VM is used for CEPH logs, other two - for the data. In total we use now 21 OSD daemons. Two host servers are running under Scientific Linux 7.2. Host servers are located in different sites with a distance about 40 Km. They are connected over Internet with data link of nominal speed 1 Gbit. Three OSD daemons are running in docker environment. Docker environment for OSD daemon is considered as most effective solution for storage server. We plan to create first open source and working version for the GDSDS next year. Of course we need support and volunteers.

7. Conclusion

We have analyzed a range of distributed storage systems, paid attention to most important features of these systems and suggested developing Geographically Distributed Software Defined Storage.

Acknowledgement

The research has been carried out with the financial support of the Ministry of Education and Science of the Russian Federation under grant agreement #14.575.21.0058 (RFMEFI57514X0058).

References

[1] Analysis of Six Distributed File Systems *HAL-Inria*, 2013 https://hal.inria.fr/hal-00789086/file/a_survey_of_dfs.pdf.

[2] Blomer J. Survey of distributed file system technology *ACAT* 2014, Prague (in references) Also iopscience.iop.org/article/10.1088/1742-6596/664/4/042004/pdf

[3] Shirinbab S. et al Performance Evaluation of Distributed Storage Systems for Cloud Computing *IJCA*, Vol. 20, No. 4, Dec. 2013, pp 195-207.

[4] Comparison of distributed file systems [Electronic resource]: https://en.wikipedia.org/wiki/Comparison_of_distributed_file_systems.

[5] Espinal X. et al Disk storage at CERN: Handling LHC data and beyond *20th International Conference on Computing in High Energy and Nuclear Physics (CHEP2013) IOP Publishing Journal of Physics: Conference Series 513* (2014) 042017 doi:10.1088/1742-6596/513/4/042017 http://iopscience.iop.org/article/10.1088/1742-6596/513/4/042017/meta

[6] Roblitz T. Towards Implementing Virtual Data Infrastructures – a case study with iRODS *Computer Science* 13 (4) 2012 http://dx.doi.org/10.7494/csci.2012.13.4.21

[7] Software Defined Storage LizardFS is a distributed, scalable, fault-tolerant and highly available file system [Electronic resource]: https://lizardfs.com/about-lizardfs/

[8] Adamas, J Why so Sirius? Ceph backed storage at the RAL Tier-1 [Electronic resource]: https://indico.cern.ch/event/466991/contributions/2136880/contribution.pdf

[9] XtreemFS is a fault-tolerant distributed file system for all storage needs [Electronic resource]: http://www.xtreemfs.org/

[10] http://www.orangefs.org/ project

[11] Shan McKee presentation: "Using Cephvia OSiRIS for ATLAS. Distributed Cephand Software Defined Networking for Multi-Institutional Research" [electronic resource] https://indico.cern.ch/event/523410/contributions/2355721/attachments/1368439/2074169/OSiRIS-ATLAS-XROOTd-workshop.pdf

[12] Wong M.-T. et al Ceph as WAN Filesystem – Performance and Feasibility Study through Simulation *Network Research Workshop Proceedings of the Asia-Pacific Advanced Network* 2014 v. 38, p. 1-11. http://dx.doi.org/10.7125/APAN.38.1 ISSN 2227-3026.

[13] Zhang X. et al Ceph Distributed File System Benchmarks on an Openstack Cloud *Conference Paper, November* 2015 https://www.researchgate.net/publication/286622938

[14] Cui L. et al When big data meets software-defined networking: SDN for big data and big data for SDN *IEEE Network*, Vol.30 Issue 1, pp 58 - 65 DOI: 10.1109/MNET.2016.7389832

[15] Xin Du et al ProSy: A similarity based inline deduplication system for primary storage *Networking, Architecture and Storage (NAS)*, 2015 IEEE International Conference on Date of Conference: 6-7 Aug. 2015 DOI: 10.1109/NAS.2015.7255230

[16] Avani Wildani et al HANDS: A heuristically arranged non-backup in-line deduplication system *Data Engineering (ICDE)*, 2013 IEEE 29th International Conference on Date of Conference: 8-12 April 2013 DOI: 10.1109/ICDE.2013.6544846

[17] Srinivasan K. et al iDedup: Latency-aware, inline data deduplication for primary storage *USENIX FAST-2012* http://dl.acm.org/citation.cfm?id=2208485 Published in Proceeding FAST'12 Proceedings of the 10th USENIX conference on File and Storage Technologies.

[18] Padamvathi V. et al Quantum Cryptography and Quantum Key Distribution Protocols: A Survey *Advanced Computing (IACC)*, 2016 IEEE 6th International Conference on Date of Conference: 27-28 Feb. 2016 DOI: 10.1109/IACC.2016.109 http://ieeexplore.ieee.org/document/7544898/

[19] Editors: Richard Gerber et al DOE High Performance Computing Operational Review (HPCOR), Enabling Data-Driven Scientific Discovery at DOE HPC Facilities June 18-19, 2014 Oakland, CA 55 pages.

[20] Jian Peng et al Simulating the Burst Buffer Storage Architecture on an IBM BlueGene/Q Supercomputer sc16.supercomputing.org/sc-archive/tech_poster/poster_files/ post143s2-file3.pdf Super Computing 2016 2 pages.

[21] Ning Liu et al On the Role of Burst Buffers in Leadership-Class Storage Systems *Mass Storage Systems and Technologies (MSST)*, 2012 IEEE 28th Symposium on Date of Conference: 16-20 April 2012 DOI:10.1109/MSST.2012.6232369 http://ieeexplore.ieee.org/document/6232369/ 11 pages.

[22] Gilbert, S., Lynch, *N.* Brewer's conjecture and the feasibility of consistent, available, and partition-tolerant web services. *ACM SIGACT News* 33, 2 (2002).