Testing the limits of an LVS - GridFTP cluster as a replacement for BeSTMan

E Fajardo[1], C Pottberg[1], B Bockelman[2], G Attebury[2], T Martin[1] and F Würthwein[1]

[1] University of California San Diego, La Jolla, CA, USA
[2] University of Nebraska, Lincoln, NE, USA

E-mail: emfajard@ucsd.edu

Abstract. The Worldwide LHC Computing Grid (WLCG) is the largest grid computing infrastructure in the world pooling the resources of 170 computing centers (sites). One advantage of grid computing is that multiple copies of data can be distributed across different sites, allowing user access that is independent of geographic location or software. Each site is able to communicate using software stacks collectively referred to as "middleware". One key middleware piece is the storage element (SE), which provides remote POSIX-like access to a site's storage.

The middleware stack managed by the Open Science Grid (OSG) used a storage resource manager (SRM) protocol implementation that, among other things, allowed sites to load-balance servers providing the Grid File Transfer Protocol (GridFTP) interface. OSG is eliminating the use of an SRM entirely and is transitioning to a solution based solely on GridFTP load-balanced at the network level with Linux Virtual Server (LVS). LVS is a core component of the Linux kernel, so this change increases both maintainability and reduces complexity of the site. In this document, we outline our methodologies and results from the large scale testing of an LVS+GridFTP cluster for data reads. Additionally, we discuss potential optimizations to the cluster to maximize total throughput.

1. Introduction

The most common model for interacting with storage on the Worldwide LHC Computing Grid (WLCG) is the *storage element* (SE). In this model, the site provides a POSIX-like storage system and exposes it via a variety of remote access protocols. Clients interact directly with the storage element to transfer files in and out, and data management layers work to catalog the SEs contents and orchestrate inter-SE transfers.

The storage element model provides rich functionality (at the cost of complexity compared to, for example a cache-based model) and a common implementation involves providing a service that speaks the *storage resource management* (SRM) protocol. SRM is used to provide metadata services, manage storage allocations, and load-balance transfer protocols such as GridFTP. Due to a variety of reasons not explored here, the former use cases have fallen out of use, leaving only load-balancing.

A niche protocol such as SRM is not needed to load-balance network services; various techniques are well-established. In this paper, we explore the use of one system, Linux Virtual Server (LVS), to load balance data services provided by a number of sites in the US.

2. Current storage architecture

2.1. GridFTP

A common need for a computing facility on the grid is to expose their storage system to external grid clients. The Globus implementation [1] of the Grid File Transfer Protocol (GridFTP) over two parties solves this problem and it also provides authentication and authorization. A single GridFTP server transfer throughput can scale up to the limitation of its own hardware network capabilities. Hence for sites that have more bandwidth available, more than one server its needed to avoid bottlenecking.

2.2. BestMan

The Worldwide LHC Computing Grid (WLCG)[2] adopted the user of Storage Resource Manager (SRM) specification for data transfers between sites and also for storing the output of computing jobs at remote sites. One of the implementations of the SRM is the Berkeley Storage Manager(BeStMan) [3] developed at LBNL (Lawrence Berkeley Nationa Laboratory) and the suggested SRM solution to sites by the Open Science Grid (OSG) [4].

BeStMan also provided the functionality of load balancing (several GridFTP servers), authentication and authorization. Hence it allowed a site to fully use its available bandwidth (by using several GridFTP servers)while only advertising a single BeStMan server.

On 2012 LBNL dropped the support for BeStMan on the OSG, leaving OSG to maintaining its codebase and providing security and performance updates. While OSG has focused on simplifying and removing features, this software product is still over 150,000 lines of code - providing motivation for finding a simpler setup.

2.3. Hadoop File System

The Hadoop File System (HDFS) [5] is a distributed file system that works in commodity, heterogeneous hardware and can store tens of PetaBytes. HDFS features automated data replication (usually in a factor of 3) and recovery of lost or corrupted files. Since 2009 several sites in the OSG use HDFS as their file storage solution [6].

The architecture of HDFS is composed of a name node (NN) and several data nodes (DN). The name node has all the meta information about files in the system, while data nodes are where the files actually reside. The interface between the GridFTP servers and the name node can happen via a FUSE mount in the GridFTP server that offer a POSIX interface to the HDFS namespace. Or it can happen through the the GridFTP HDFS plug-in which converts GridFTP requests into HDFS requests using the HDFS libraries. The latter is the preferred and best performance solution.

3. LVS

LVS is a Linux Kernel based load balancing system project. The project combines the IP Virtual Server (IPVS) code that has been present in the Linux kernel since 2.4 with a variety of user level daemons and tools eg. Keepalived (http://www.keepalived.org).

3.1. IPVS

IPVS uses Layer 4, or transport layer, switching to provide load and reliability balancing for a cluster of Real Servers while providing a single IPv4(v6) Virtual Server address for access.

3.2. Keepalived

IPVS uses features of the kernel in combination with the daemon Keepalived. Keepalived uses the VRRP protocol and series of checks to dynamically manage the pool of real servers based on their health. It then directs IPVS to perform the required routing to have the Real Servers respond.

3.3. Direct Routing
At UCSD we use the LVS in the direct routing mode. This mode is used when the both the real servers, and the Virtual Servers have real world routable IPv4(v6) network address and can be directly accessed over the world wide network. Other models are available in LVS eg. Nat (but they were not tested for this work).

3.4. Linux Networking Customizations
In LVS the Director, which has the virtual server address, forwards all TCP/IP requests it gets to the correct real server after performing various book keeping duties (eg. tracking connections). The mechanisms on how this works depends if the underlaying network is IPv4 or IPv6.

3.4.1. IPv4 For IPv4 there are two key mechanisms to make this work. The interface on the Virtual Server and all real servers are given the same IPv4 network address. This creates a problem where multiple servers are responding to the same IPv4 address via the AARP protocol. Arptables is then used to filter these responses so that only the Virtual Server receives the corresponding ARP responses from the real Servers.

3.4.2. IPv6 In addition to the ARP filtering with IPv4 the link local interface assigned to the Virtual Server is the same as the one of the Real Servers. This allows the real server to only respond to requests on the local network for those requests unless the director has decided otherwise.

3.5. GridFTP Protocol and IPVS
The GridFTP protocol is similar to the FTP protocol in that it uses both a control and a data channel. The control channel operates to create the initial connection and to establish the data channels. As is most often the case the data channel operates in passive mode. This means that the GridFTP server starts up a process that listens in a port in a set range. The client then connects to that port and starts the data transfer. However this creates a problem since the data channel port is random and assigned after the Director has forwarded the control channel request.

For GridFTP the the protocol allows a workaround for this problem. The GridFTP server can be configured to tell the client to connect to a specific hostname for the data channel. By setting this configuration variable (data_interface) to the Real Servers hostname the client can be directed to directly connect to the data port on the GridFTP server selected by the LVS Keepalived.

4. LVS and GridFTP scalability
Scalability tests were performed to confirm that LVS was a good enough replacement for BeStMan . In this section we describe the procedures and the results of the scaling exercises that were performed.

4.1. Test procedure
The objective of the test was to generate a load of a given number of clients requesting from the LVS server and measure the total throughput. GlideinWMS (Glide-in based Workload Management System)[7] was used to submit several jobs (clients) at the same time to requests files through the LVS. Each job consisted of a cycle of submissions requesting a file at random, sleeping for a given time and then requesting another one.

In order to obtain the most number of distributed cores available and to do so without impacting production operations in the cluster a sleeper pool was used. A sleeper pool consist

of a batch system pool in which compute nodes are configured to advertise more cores than what they have, under the premise that those extra slots will consume a negligible amount of resources(CPU, memory and disk).

All the jobs in these tests used the *gfal-copy* command line utility to request the files from the LVS server and dumped them into */dev/null*. To prevent HDFS from being a bottleneck and isolate the performance of the LVS setup, the set of files for the test were guaranteed to have a replication factor of ten in Hadoop.

4.2. LVS Directors and GridFTP Server Hardware Setup

The two LVS redirectors, and 6 GridFTP servers at UCSD run on identical single CPU blade servers.

(i) Intel(R) Xeon(R) CPU E5-2609 v4 @ 1.70GHz
(ii) 64GB RAM
(iii) Intel Corporation 82599 10 Gigabit Network Interface
(iv) Two 80GB SSD Drives

4.3. Results

We measured the total throughput of the LVS system at different concurrencies (parallel running jobs) and found out that the throughput was correlated with the number of GridFTP clients, see Figures 1 and 2, which is consistent with the results from [6]. The average and maximum total throughput are summarized in Table 1. It is worth noticing than although done with different hardware (but same network capacity) the maximum throughput achieved is more than twice than from the previous result in [6]. Finally we fit in Figure 3 the average throughput at each level of concurrency and can see that although the throughput scales with the number of clients as previously noted in [6], it is only true up to one point when the the total performance starts to decrease.

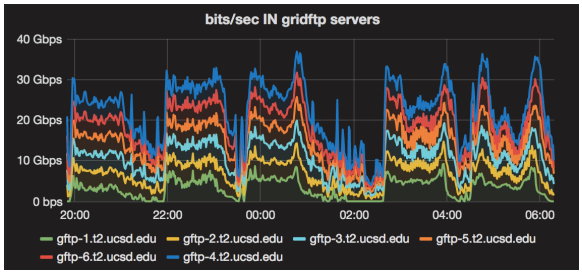

Figure 1. Combined throughput with different number of GridFTP clients

Figure 2. Number of running parallel jobs.

5. Conclusions

In this paper we show how the architecture of the Linux Virtual System in combination for the Globus Grid File transfer Protocol meets the needs of replacing BeStMan as a SE solution for OSG grid sites: load balancing a set of GridFTP servers while providing a single grid entry point to a site's storage. Moreover its bandwidth throughput is equal or higher than the one provided by BeStMan , while at the same time offering high availability, IPV6 compatibility and a single advertising point. There are still some problems in the future as more sites adopt this since installation and configuration of LVS requires more expertise than the BeStMan one.

Parallel clients	Average Throughput $Gbit/s$	Maximum Throughput $Gbit/s$
1000	22.945	32.934
2000	26.853	34.211
3000	28.384	37.190
4000	26.263	36.328
5000	23.334	37.076

Table 1. Average and maximum throughput at different concurrency levels

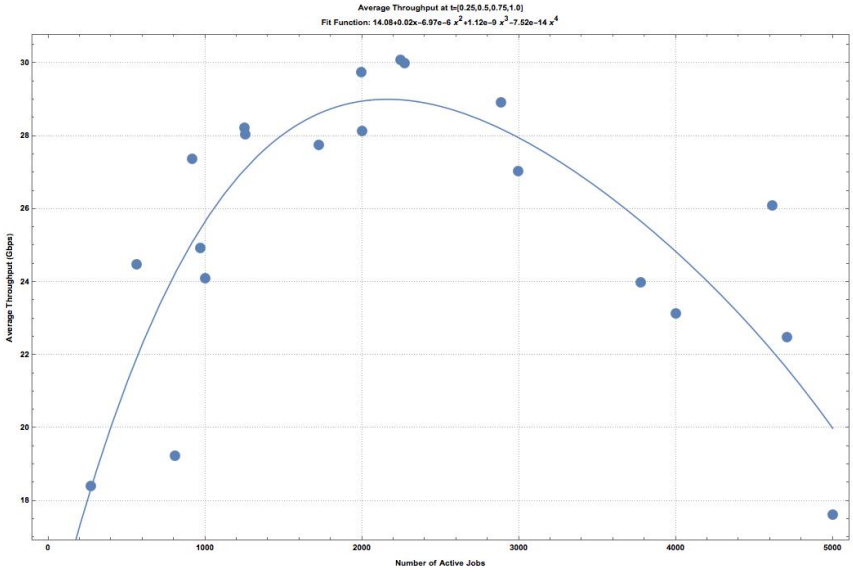

Figure 3. Fit for the average throughput at different concurrency levels

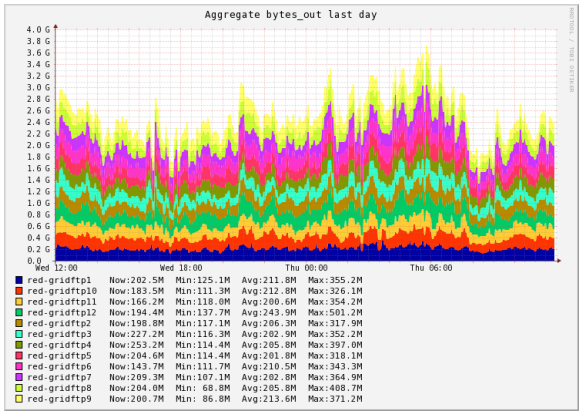

Figure 4. Tipical throughout of gridftp servers at UNL

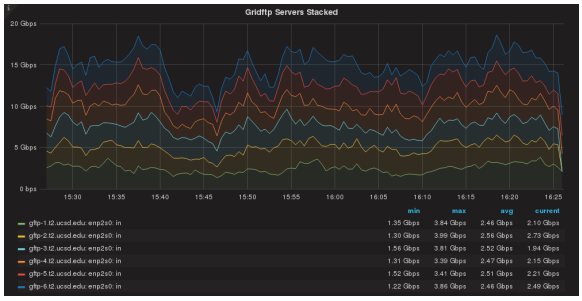

Figure 5. Tipical throughout of GridFTP servers at UCSD

However for sites like Nebraska and UCSD which have migrated, they have seen good results, see figures 4 and 5.

Looking into the future, similar tests should be performed once Hadoop 3 is available to

see if there is an actual difference in performance. Finally, although reasonable maximum throughputs were achieved in this test, they did not achieve the maximum physical installed bandwidth ($60Gbit/s$) of the site (six GridFTP servers with $10Gbit/sec$ network interfaces) . We believe this is mostly caused because of the way sleeper pool works, where to many jobs can land in a single computing host and are throttled by the $1Gbit/sec$ network interfaces.

Acknowledgments
This work is supported in part by the National Science Foundation through awards PHY-1148698 and ACI-1321762.

References

[1] Allcock W, Bresnahan J, Kettimuthu R, Link M, Dumitrescu C, Raicu I and Foster I 2005 The globus striped gridftp framework and server *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing* SC '05 (Washington, DC, USA: IEEE Computer Society) pp 54– ISBN 1-59593-061-2 URL `https://doi.org/10.1109/SC.2005.72`

[2] Bird I 2011 Computing for the large hadron collider vol 61 (Annual Reviews) pp 99–118 URL `http://www.annualreviews.org/doi/abs/10.1146/annurev-nucl-102010-130059`

[3] Berkeley storage manager (bestman) URL `https://sdm.lbl.gov/bestman/`

[4] Pordes R, Petravick D, Kramer B, Olson D, Livny M, Roy A, Avery P, Blackburn K, Wenaus T, Wrthwein F, Foster I, Gardner R, Wilde M, Blatecky A, McGee J and Quick R 2007 *Journal of Physics: Conference Series* **78** 012057 URL `http://stacks.iop.org/1742-6596/78/i=1/a=012057`

[5] Attebury G, Baranovski A, Bloom K, Bockelman B, Kcira D, Letts J, Levshina T, Lundestedt C, Martin T, Maier W, Pi H, Rana A, Sfiligoi I, Sim A, Thomas M and Wuerthwein F 2009 Hadoop distributed file system for the grid *2009 IEEE Nuclear Science Symposium Conference Record (NSS/MIC)* pp 1056–1061 ISSN 1082-3654

[6] Amin A, Bockelman B, Letts J, Levshina T, Martin T, Pi H, Sfiligoi I, Thomas M and Wüerthwein F 2011 *Journal of Physics: Conference Series* **331** 052016 URL `http://stacks.iop.org/1742-6596/331/i=5/a=052016`

[7] Sfiligoi I, Bradley D C, Holzman B, Mhashilkar P, Padhi S and Wurthwein F 2009 The pilot way to grid resources using glideinwms *2009 WRI World Congress on Computer Science and Information Engineering* vol 2 pp 428–432