High performance data transfer

IOP Conf. Series: Journal of Physics: Conf. Series **898** (2017) 062025 doi:10.1088/1742-6596/898/6/062025

High performance data transfer

R Cottrell[1], C Fang[2], A Hanushevsky[1], W Kreuger[1] and W Yang[1]

[1] SLAC National Accelerator Laboratory, Menlo Park, CA, US

2. Zettar, Mountain View, CA, US

E-mail: cottrell@slac.stanford.edu

Abstract. The exponentially increasing need for high speed data transfer is driven by big data, and cloud computing together with the needs of data intensive science, High Performance Computing (HPC), defense, the oil and gas industry etc. We report on the Zettar ZX software. This has been developed since 2013 to meet these growing needs by providing high performance data transfer and encryption in a scalable, balanced, easy to deploy and use way while minimizing power and space utilization. In collaboration with several commercial vendors, Proofs of Concept (PoC) consisting of clusters have been put together using off-the-shelf components to test the ZX scalability and ability to balance services using multiple cores, and links. The PoCs are based on SSD flash storage that is managed by a parallel file system. Each cluster occupies 4 rack units. Using the PoCs, between clusters we have achieved almost 200Gbps memory to memory over two 100Gbps links, and 70Gbps parallel file to parallel file with encryption over a 5000 mile 100Gbps link.

1. Introduction

We report on the results of a collaboration between a start-up (Zettar) and the SLAC National Accelerator Center (SLAC) to provide state of the art, efficient, cost-effective, scalable high speed data transfers over Wide Area Networks (WANs). The exponential growth in the needs for fast reliable transfer of massive amounts of data over large distances has become increasingly critical. This is required, for example, by: Defense; the Oil and Gas, Media and Entertainment industries; HPC; and Life and modern Distributed Data Intensive Sciences. In this paper we focus on a single requirement example: the existing and next generation Free Electron Laser (FEL) being constructed at SLAC in Menlo Park California The concept is extensible to the other areas mentioned above.

2. Requirements

There is an increasing need to efficiently copy data from one location where it is created to another where it can be analyzed. For example, by 2024, the world-leading Linear Coherent Light Source (LCLS) FEL will require the ability to transfer files at one Terabit/second from SLAC to leadership class computers [1] at, for example, the National Energy Research Scientific Center (NERSC) at the Lawrence Berkeley National Laboratory (LBNL).

Figure 1 shows the growth in requirements for LCLS/LCLS-II now and in the near future together with what might be expected from Moore's law improvements and what has been achieved so far with Zettar's PoC's.

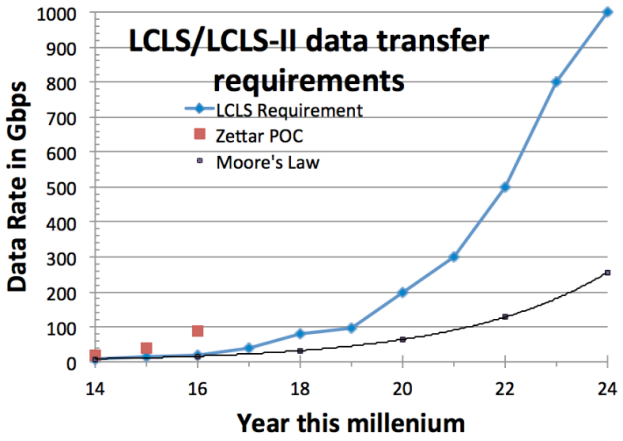

Figure 1: LCLS-II file transfer throughput requirements.

It is seen that we cannot expect improvements following Moore's Law [2] on its own to meet the requirements. We therefore need to come up with a modern balanced, cost effective system design that scales in multiple dimensions including: network interfaces (links and Network interface Cards (NICs); CPUs and cores; and Input/output Operations Per Second (IOPS). At the same time we need to carefully select hardware to take optimal advantage of technologies such as clusters and Solid State Drive (SSD) memory.

3. Goals

As mentioned above we focused on the LCLS-II requirements to transfer data from SLAC to NERSC over existing and future expected links with high bandwidth, high link quality and low competing traffic such as those provided today by the US DoE's ESnet. We settled on the following goals:

- High availability using peer-to-peer software with clusters, and with automatic failover incorporating multiple CPUs and NICs.
- Scale out by utilizing 1 Rack Unit (RU) devices that can easily be replicated.
- Low component count, low complexity and as far as possible everything managed by software.
- Forward looking, e.g. today utilizing 100Gbps Infiniband (IB) EDR Host Channel Adapters (HDAs) and 25Gbps NICs.
- Storage tiering friendly including a hierarchy of Data Transfer Node (DTN) through SSDs to rotating storage.
- Cost effective: e.g. careful balancing by using top of the line SSDs for writing, while utilizing less expensive SSDs for reading.
- Small form factor for the base system.
- Energy efficient

4. Design

There are two clusters, typically a sender and a receiver. The design is software based. Under the guidance of a "data transfer controller" peer, the peers in the cluster work together to slice a data set, regardless of datatypes (numerous small files, large files, or a mix) into an appropriate number of "logical pieces" to be transferred to a destination over multiple NICs and multiple streams.

A single cluster for the latest Zettar PoC is shown in Figure 2.

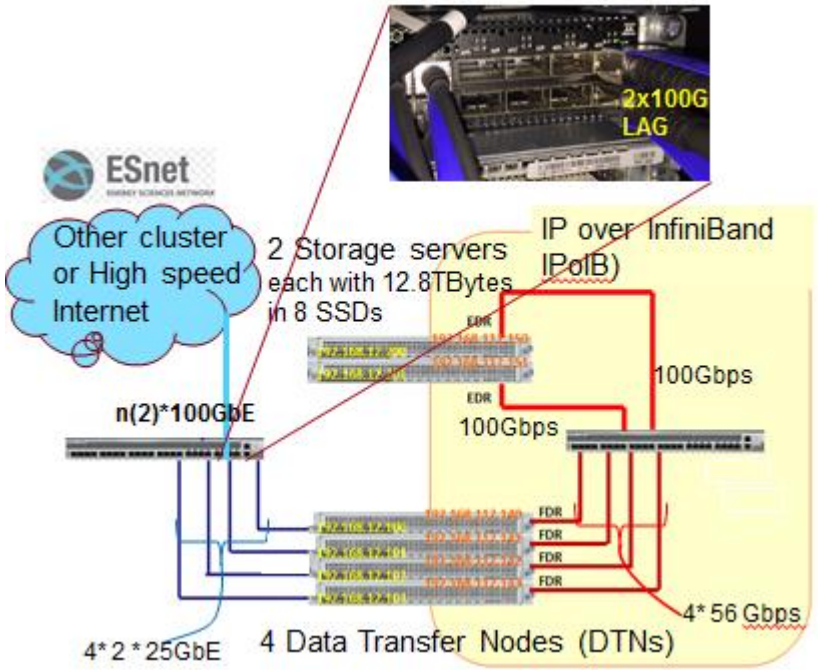

Figure 2: Zettar's Third generation Proof of Concept cluster

Shown in the Figure are

- Two 1RU storage servers each holding 12.8TBytes in eight 1.6TByte SSDs. They are running the BeeGFS [3] parallel file system and connected by two 100Gbps EDR IB links to an IB switch.
  - We utilize 2*1RU servers rather than say a four node 2RU cluster to assist in scalability.
  - We utilize IB rather than Ethernet for the local links since it is simpler to setup and less tuning is needed.
    - The downside is we need a second (IB) switch in addition to the Ethernet switch.
- The IB switch in turn connects via four 56Gbps FDR IB links to four 1RU DTNs.
  - Each DTN has two 25Gbps NICs and two Intel E5-2620v3.2 2.4 GHz CPUs.
- Each of the four DTNs in turn connects to an Ethernet switch with two 25GbE links.
- The DTNs and storage servers are all running Linux (CentOS 7.2 x86_64).
- The Ethernet switch, under the control of an administrator, can connect to either:
  - Directly to a second identical cluster in the same rack using 2*100Gbps Link Aggregated Group (LAG) NICs,
  - Or via a 100Gbps link to a 10Gbps border router port that in turn connects to an ESnet link to go offsite.
- The ESnet connection is nominally a shared 100Gbps link, but it is controlled by OSCARS [4] to guarantee delivery of a known bandwidth and allow going above this if the bandwidth is available.
  - The ESnet link is routed via Atlanta and back (~ 5,000 miles) from whence it goes to the second identical cluster in the same rack.
- The use of two identical clusters in the same rack greatly simplified the installation, setup and debugging of problems.

5. Results

5.1. Memory to memory transfers

Using the two 100Gbps links we transferred memory to memory data between the DTNs in the two locally connected clusters separated by a few metres. The operating system had TCP with Fast Open enabled [5] but this is not a critical requirement. The results for three transfers of 1TByte each are shown below in Figure 3.

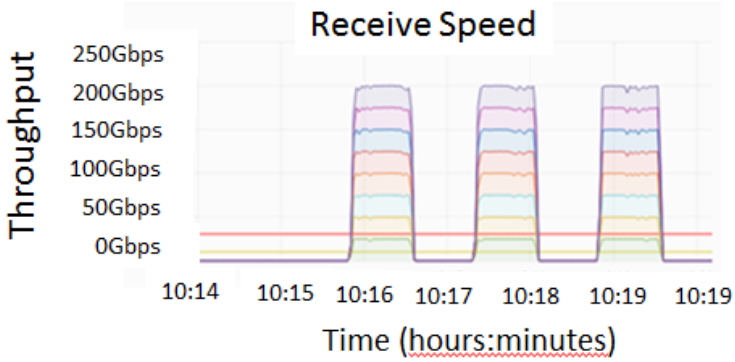

Figure 3: Memory to memory transfer over two 100Gbps links. The bottom axis is the time in hours:minutes. Each layer represents the throughput of each 25Gbps interface.

Note the uniformity of the 8*25Gbps interfaces and that in aggregate we achieved close to the lien rate 200Gbps. We also tested the memory to memory throughput while injecting latency with little change. From this we conclude that for up to 200Gbps on clean networks, the network is not the problem and we do not need the complexity of non-TCP based protocols.

5.2. File to file transfers

For file to file transfers we introduce more parameters that impact performance. These include: the raw read and write speeds of the SSDs; the base file system (XFS [6] in our case); and the parallel file system (BeeGFS).

Using XFS just to read the data in a file server and measuring the performance (i.e. measured inbox in a storage server directly) using the Unix *fio* utility we get the result show in Figure 4.

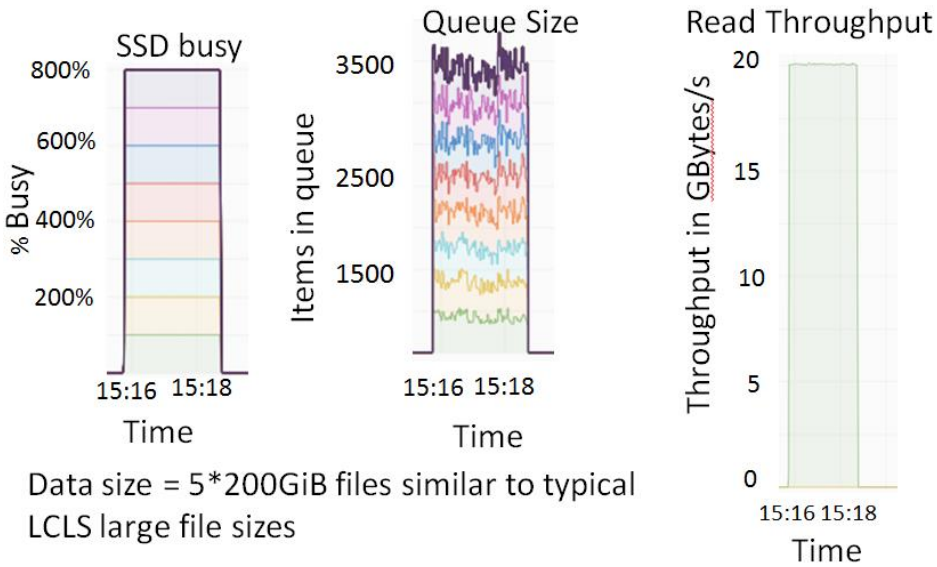

Figure 4: XFS read performance of 8*SSDs in a file server measured using the Unix *fio* utility

It is seen that the SSDs are uniformly busy and the queues are kept full resulting in close to 20 GBytes/s data read speed.

However, when we look at the write performance including XFS and the parallel file system using 16*SSDs in two file servers, measured in the storage clients performance drops dramatically as seen in Figure 5. Note that the BeeGFS parallel file system aggregates XFS file systems. It is of interest to note that logically, measuring the available storage IOPS and throughput should always be made in a storage client. Inbox measurements should always be used as baselines.

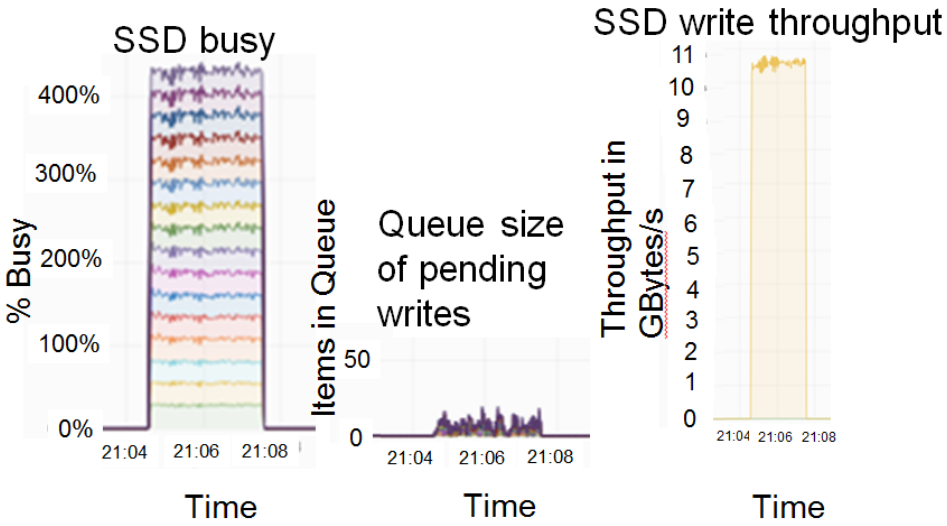

Figure 5: Write performance of file servers.

The write performance is seen to be ~11GBytes/s or almost a factor of 2 slower than the read performance. A major cause is the inability of keeping the queue size of pending writes full. Its size is a factor of 1000 times fewer items queued for writes than reads.
Figure 6 shows the impact on performance of the various layers involved.

SSD speeds	Read	Write	Write/ Read	% Intel Read	% Intel Write	IOPS read	IOPS write	IOPS write / read
Intel sequential	2.8 GBps	1.9 GBps	68%	100%	100%	1200	305	25%
XFS	2.4 GBps	1.5 GBps	63%	86%	79%			
XFS/Intel	85%	79%						

Raw Intel sequential write speed 68% of read speed

Limited by IOPS

XFS 79%(write)-85%(Read) of Raw Intel sequential speed

File speeds	Read	Write	Write/Read
XFS	38GBps	24GBps	63%
BeeGFS+XFS	21GBps	12GBps	57%
BeeGFS+XFS/XFS	55%	50%	

Figure 6: Breakdown of the impacts of various layers on performance. Note GBps = GBytes/s versus Gbps=Gbits/s.

Comparing the sequential performance of the SSDs to that achievable with XFS and the parallel file system we see that:

- Comparing sequential SSD write vs. read performance, write is 65% of read performance.
- IOPS for writes is only 25% of IOPS for read.
- XFS achieves 85% (read) and 79% (write) of the sequential SSD performance.
- The parallel file system (BeeGFS) further impacts performance so that the parallel file system has a throughput of only 55% (read) and 50% (write) of XFS.
- The net reductions of the parallel file system is 47% (read) and 38% (write) of the sequential SSD performance.

5.3. Encryption

In 2015, using an earlier PoC [7] without the employment of external storage servers (i.e. all NVMe SSDs were installed in DTNs, and BeeGFS ran directly on DTNs) we tested the effects of Transport Layer Security (TLS [8]) encryption on file to file transfers utilizing the 5,000 mile 100Gbps link. The TLS encryption used X.509 certificates and Perfect Forward Security (PFS [9]). CPU assisted hardware encryption acceleration was not used. The employed test data set consisted of files of the same size, e.g. 20,000x50MiB, 2,000x500MiB, 400x5GiB etc. They showed transfer speeds of over 70Gbps. The one shown below in Figure 7, 20,000x50MiB, has an average speed in the middle of the pack. For TLS, the employed ciphersuite was `RSA_WITH_AES_128_GCM_SHA256`. For PFS, the employed ciphersuite was `ECDHE_RSA_AES_128_GCM_SHA256`.

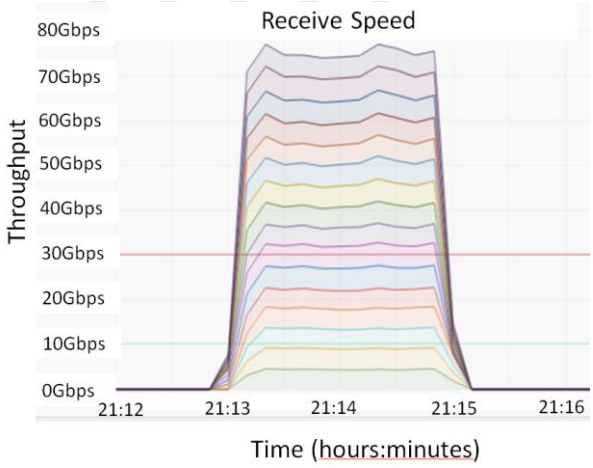

Figure 7: Data transfer speed profile time histories for the receiving side. Test data set employed consists of 20,000x50MiB files with TLS encryption.

Note the even utilization of all 16 NICs, and also note no interface bonding was needed. Basically we observed that TLS encryption degraded the throughput by ~15%.

5.4. Lots of Small Files (LoSF)

Typical data sets consist of not only large-size file but LoSF can be a challenge since it takes more resources to manage all the files. For example in [10] the authors reported that for LOSF, GridFTP only attained around 4Gbps over a 100Gbps network. Over the 5000 mile link to Atlanta and back with a test set of 1,000,000 (~1 TiB) transferred with TLS encryption, the transfer speed is ~30Gbps, see Figure 8.

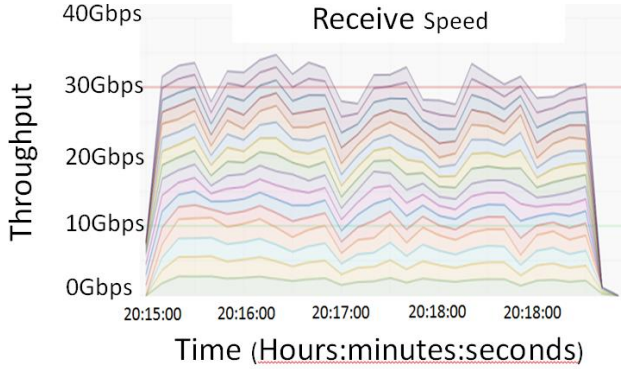

Figure 8: Data transfer speed profile time histories for the receiving side for LoSF. The test data set employed consists of 1,000,000x1MiB files. Note the even utilization of all 16 network interfaces.

6. Conclusions

- The ZX software employs asynchronous processing, not just multiple threads in each instance. This approach increases its use of the available computing resources (storage, compute, and network). It unconditionally uses multiple streams (the number depends on the file sizes and other run time conditions) over multiple NICs. Due to its use of short-duration TCP streams, the kernel's network stack doesn't need extensive tuning. Although over the WAN, some techniques documented in fasterdata.es.net [11] for host tuning could be helpful (e.g. the use of fair queuing), but not essential. The software is designed with a co-design approach. The use of a cluster also is essential given today's operating systems' still primitive ability of using multiple CPU sockets and the typical motherboard designs. So, to use it more effectively, the data transfer system design should be matching. A reference design has been documented in [12] and further elaborated in [7]. Today, data transfers at rates higher than 10Gbps require co-design such that storage, compute, network, and software are considered in an integrated manner to achieve near optimal efficiency of a computing task (e.g. a data transfer)
- The bottleneck today over clean networks does not appear to be the network. We can drive 200Gbps utilizing TCP and there is no need for proprietary protocols, allowing us to easily take advantage of improvements being made to TCP performance. Admittedly our demonstration was over a short link, since currently, we do not have access to a long distance 200Gbps link. However, with the use of multiple NICs, multiple streams and clusters we believe we can sustain this rate over the longer distances.
- The main performance bottlenecks today are:
    - Insufficient write IOPS
        - It is important to use high performance SSDs. We used Intel DC P3700 NVMe 1.6TByte drives.
            - Unfortunately the fastest SSDs are also the most expensive with typically a 20% improvement for a 60% cost increase.
    - The parallel file system.
- Future work includes:
    - Exploring how to improve or avoid the parallel file system degradation in performance.
    - Better integration with hierarchical storage management.
    - Tracking and taking advantage of technology improvements.
    - Getting access to faster non-local-area networks for testing.

Acknowledgements
We had valuable discussions with Brent Draney and Jason Lee of NERSC, Chin Guok and Brian Tierney of ESnet, Amedeo Perazzo of SLAC/LCLS, Guillaume Cessieux of SLAC, Antonio Ceseracciu then of SLAC now at eBay. We acknowledge the assistance of Ron Barrett and John Weisskopf of SLAC for installing the equipment. We are extremely grateful to the many vendors who provided advice and evaluation equipment. These include: AIC, Arista, Cisco, Dell, Intel, Mellanox, and QCT. In addition we thank BeeGFS for making the parallel file system available and providing advice and upgrades. We also acknowledge the continued encouragement of James Williams and Theresa Bamrick of SLAC.

Bibliography

[1] J. Thayer and A. Perazzo, *Exascale Requirements LCLS II Case Study,* to be published, 2016.

[2] Wikipedia, "Moore's law," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Moore's_law. [Accessed 23 3 2017].

[3] BeeGFS, "BeeGFS - The Parallel Cluster File System," [Online]. Available:

http://www.beegfs.com/content/. [Accessed 10 2 2017].

[4] ESnet, "OSCARS On-Demand Secure Circuits and Reservation System," [Online]. Available: https://www.es.net/engineering-services/oscars/. [Accessed 10 2 2017].

[5] Wikipedia, "TCP Fast Open," [Online]. Available: https://en.wikipedia.org/wiki/TCP_Fast_Open. [Accessed 10 2 2017].

[6] Wikipedia, "XFS," [Online]. Available: https://en.wikipedia.org/wiki/XFS. [Accessed 10 2 2017].

[7] C. Fang, "High Performance Data Transfer for Distributed Data Intensive Sciences," SLAC Technical Note: SLAC-TN-16-001, 2016.

[8] Wikipedia, "Transport Layer Security," [Online]. Available: https://en.wikipedia.org/wiki/Transport_Layer_Security. [Accessed 24 03 2017].

[9] Wikipedia, "Forward Security," [Online]. Available: https://en.wikipedia.org/wiki/Forward_secrecy. [Accessed 24 03 2017].

[10] A Rajendram, P Mhashilkar, H Kim, D Dykstra, G Garzoglia and I Raicu, "Optimizing Large Data Transfers over 100Gbps Wide Area Networks," *13th IEEE/ACM International Symposium on Grid and Cloud Computing,,* 2013.

[11] B. Tierney, "ESnet Fasterdata Knowledge Base," ESnet, [Online]. Available: http://fasterdata.es.net/. [Accessed 25 03 2017].

[12] C. Fang, "Using NVMe Gen3 PCIe SSD Cards in High-Density Servers for High-performnace Big Data Transfer over Multiple Network Channels," SLAC-TN-15-110, SLAC National Accelerator Laboratory, Menlo Park, 2015.