High Performance Data Transfer for Distributed Data Intensive Sciences

Chin Fang
Zettar Inc., Mountain View, California, USA

R. "Les" A. Cottrell, Andrew B. Hanushevsky,
Wilko Kroeger, Wei Yang
SLAC National Accelerator Laboratory
2575 Sand Hill Road, Menlo Park, California, USA

Abstract — We report on the development of ZX software providing high performance data transfer and encryption. The design scales in: computation power, network interfaces, and IOPS while carefully balancing the available resources. Two U.S. patent-pending algorithms help tackle data sets containing lots of small files and very large files, and provide insensitivity to network latency. It has a cluster-oriented architecture, using peer-to-peer technologies to ease deployment, operation, usage, and resource discovery. Its unique optimizations enable effective use of flash memory. Using a pair of existing data transfer nodes at SLAC and NERSC, we compared its performance to that of bbcp and GridFTP and determined that they were comparable. With a proof of concept created using two four-node clusters with multiple distributed multi-core CPUs, network interfaces and flash memory, we achieved 155Gbps memory-to-memory over a 2x100Gbps link aggregated channel and 70Gbps file-to-file with encryption over a 5000 mile 100Gbps link.

Keywords — *component; data transfer; high data rates; network; cluster; flash memory; scalable*

SLAC National Accelerator Laboratory, Stanford University, Stanford, CA 94309

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-76SF00515 and Citrix Systems, Inc.

High Performance Data Transfer for Distributed Data Intensive Sciences*

Chin Fang
Zettar Inc
Mountain View, California, USA

R. "Les" A. Cottrell, Andrew B. Hanushevsky,
Wilko Kroeger, Wei Yang
SLAC National Accelerator Laboratory
2575 Sand Hill Road, Menlo Park, California, USA

Abstract — **We report on the development of ZX software providing high performance data transfer and encryption. The design scales in: computation power, network interfaces, and IOPS while carefully balancing the available resources. Two U.S. patent-pending algorithms help tackle data sets containing lots of small files and very large files, and provide insensitivity to network latency. It has a cluster-oriented architecture, using peer-to-peer technologies to ease deployment, operation, usage, and resource discovery. Its unique optimizations enable effective use of flash memory. Using a pair of existing data transfer nodes at SLAC and NERSC, we compared its performance to that of bbcp and GridFTP and determined that they were comparable. With a proof of concept created using two four-node clusters with multiple distributed multi-core CPUs, network interfaces and flash memory, we achieved 155Gbps memory-to-memory over a 2x100Gbps link aggregated channel and 70Gbps file-to-file with encryption over a 5000 mile 100Gbps link.**

Keywords—component; data transfer; high data rates; network; cluster; flash memory; scalable

I. INTRODUCTION

The exponential growth in the needs for fast reliable transfer of massive amounts of data over large distances for data intensive science has become increasingly critical. Today, SLAC's[1] Linear Coherent Light Source (LCLS) free electron laser requires 40Gbps, and the under construction LCLS-II[2] requires data rates of 800Gbps by the end of the decade and increasing to 50 Tbps by around 2025. SLAC hosts the US West Coast tier-2 center for the LHC ATLAS[3] experiment which today uses up to 35Gbps to gather data from peer sites, and anticipates an increase by a factor of ten by 2020. SLAC is also a major partner in the Large Synoptic Space Telescope (LSST[4]) in Chile that will be gathering ~2000 * 7GBytes of data per night.

In preparation, we are exploring how to meet these needs in a scalable fashion taking into account how to support and balance the needs of data management and the availability of the storage IOPS, compute power, and network bandwidth.

The paper describes the work done to develop software that addresses the above needs, and the testing to show its performance both for memory to memory and file to file with encryption, on existing production platforms plus a Proof of Concept (PoC) using two four-node clusters. Each cluster is equipped with flash storage and 16x10Gbps network interfaces, together with a 2x100Gbps link aggregated channel, as well as a 100 Gbps link over a 5000 mile path with a 120-130msec Round Trip Time (RTT).

The rest of the paper is organized as follows: Section **II** describes related work. Section **III** outlines in more detail the requirements we face. Section **IV** describes the data transfer software. Section **V** covers the testing. Section **VI** provides the performance results. Conclusions are drawn in Section **VII** together with more discussions, and future work is envisioned in Section **VIII**.

II. RELATED WORK

In this section, we would like to provide a brief review of the development of high-performance data transfer tools, both free and commercial. In the research community, the development of high performance data transfer applications started in earnest in the late 90's with the appearance of ANL's[5] GridFTP[6] and SLAC's bbcp[7] in 2001. Up until then, there were no readily available high performance transfer tools that could effectively use 100% of the available network bandwidth. The need for such tools became crucial as network speeds increased, driven largely by the web, allowing users to run data-intensive applications at locations requiring input data placement and output offloading. GridFTP used a

[1] See https://www6.slac.stanford.edu/
[2] See https://portal.slac.stanford.edu/sites/lcls_public/lcls_ii/Pages/default.aspx
[3] See https://en.wikipedia.org/wiki/ATLAS_experiment

* Work supported by U.S. Department of Energy, Office of Science, under Contract No. DE-AC02-76SF00515.

[4] See http://www.lsst.org/lsst/
[5] See http://www.anl.gov/
[6] See https://en.wikipedia.org/wiki/GridFTP
[7] See https://www.slac.stanford.edu/grp/scs/paper/chep01-7-018.pdf

client-server approach while bbcp used a then popular peer-to-peer technique. As the authors of this paper all know bbcp well, we will describe its evolution as an illustration of how a tool in this category came into being. The main motivation behind bbcp was to provide users the ability to do high performance point-to-point copies with familiar copy command syntax without administrative intervention (i.e. installing server software). Since this approach bypassed all administrative controls, bbcp also included a congestion control algorithm based on inferred packet loss to avoid overrunning the network. bbcp also turned out to be a good network test tool since it could decouple data residence (e.g. disk) from the actual network capabilities and allow independent tuning of each data transfer aspect. This was highlighted by using bbcp in the wining 2005 Supercomputing Bandwidth Challenge contest[8].

Since that time, numerous other high-performance data transfer tools have been developed - both commercial, e.g. Aspera[9], and free, e.g. FDT[10]. Many such tools are reviewed and categorized by the excellent DOE ESNet fasterdata site[11].

Note that despite being two of the earliest entries in this category, both GridFTP and bbcp remain as popular[12] choices in industry, academia, and government, and are still being actively maintained. bbcp, for example, continues to evolve as users ask for enhancements such as better firewall accommodation, small file support, and scp-like semantics.

We also wish to note that nearly all the commercial tools in this category were also created about 10 years ago. They evolve; however, their architecture, design thinking, and capabilities have not evolved sufficiently. The lack of: a balanced consideration to storage IOPS, compute power availability, network bandwidth, and the need to be scale-out capable is particularly prevalent. Thus, most of the commercial software would be hard-pressed to go beyond 10Gbps[13] even running on well-tuned Data Transfer Nodes (DTNs[14]) and high-speed networks maintained and operated by experts.

As far as we know, among all the widely used free tools, except GridFTP and XRootD[15], the rest do not include an HPC oriented cluster architecture. Given the exponential data-growth trend in this space, and the obviously limited availability of computing resources from a single host, it should evident that the gap between the demand and the capability of the majority of host-based free solutions will grow wider and wider. Furthermore, often times, data sets consist of not only large-sized files, but also Lots Of Small Files (LOSF). The latter is really challenging to today's data transfer tools and users.

For example:

- In the 2013 paper "Optimizing Large Data Transfers over 100Gbps Wide Area Networks"[1], the authors reported that for LOSF, GridFTP only attained around 4Gbps over the 100Gbps Advanced Network Initiative (ANI)[16]. In the **CONCLUSION AND FUTURE WORK** section of the cited paper, the authors also observed that the majority of the tools that they evaluated showed the same deficiency. Based on our recent testing using both Globus[17] and GridFTP between two existing DTNs at SLAC and NERSC, the situation remains unchanged.
- A data set consisting of 327680 8KiB files, total size 20GiB, took more than 11 hours to transfer from SLAC to NERSC over a 10Gbps connection with bbcp using a common approach, which is presumably the approach that most end users know about and use. Few users would study bbcp's online documentation and use a recommended approach[18], which actually can often achieve much faster transfer rates for LOSF, e.g. ~ 12 minutes for the aforementioned test data set.

In the next section, we will first discuss the data transfer requirements common to modern large-scale data-intensive science and engineering, and in the section following, we will discuss the creation of data transfer software that is designed for the future.

III. REQUIREMENTS

A. Data-intensive science

Modern data-intensive science projects are highly distributed in nature; for example, just to pick three projects in photon science, high energy physics, and astronomy:

- The LCLS project is the world's first hard X-ray free-electron laser. Its strobe-like pulses are just a few millionths of a billionth of a second long, and a billion times brighter than previous X-ray sources. It has been generating data at SLAC but must transfer much of it to NERSC[19] for analysis, leveraging the latter's super computing power availability. The data transfer needs for LCLS and the follow on LCLS-II are growing exponentially from 40 Gbps in 2016 to 800Gbps in 2020 and 50 Tbps in 2025[2].

[8] See https://www.caltech.edu/news/world-network-speed-record-shattered-third-consecutive-year-1082
[9] See http://asperasoft.com/
[10] See http://monalisa.cern.ch/FDT/
[11] See http://fasterdata.es.net/
[12] Google search "bbcp utility recommendations"
[13] See http://fasterdata.es.net/data-transfer-tools/commercial-tools/
[14] See http://fasterdata.es.net/science-dmz/DTN/
[15] See http://xrootd.org/
[16] See https://esnetupdates.wordpress.com/advanced-network-initiative-ani/
[17] See https://www.globus.org/
[18] See https://www.slac.stanford.edu/~abh/bbcp/#_Toc392015155
[19] See http://www.nersc.gov/

- The ATLAS experiment at CERN [20] probes the fundamental structure of the universe and distributes the data to over 2900 collaborators worldwide e.g. a single collaborating site at SLAC today fetches data at a rate up to 35Gbps from 15 data sources worldwide to SLAC. By 2020 the ATLAS data rates will increase ten times. Fast data transfer is important to the future of ALTAS computing. It allows ATLAS to rapidly move data opportunistically to large resources such as High Performance Computing (HPC) centers. It also permits ATLAS computing to reduce the need for cache oriented storage, which currently accounts for a large part of the Tier 2 storage.
- The Large Synoptic Space Telescope (LSST) in Chile will be able to map the entire visible sky in just a few nights. Images will be immediately analyzed to identify objects that have changed or moved. In the ten-year survey lifetime, LSST will map tens of billions of stars and galaxies. With this map, scientists will explore the structure of the Milky Way, determine the properties of dark energy and dark matter, and make discoveries that we have not yet imagined. It will take ~2000 * 7GBytes of pictures per night that must be transported to the NCSA[21] in Illinois and then distributed to partners around the world.

B. *Others*

Many large and established industries are data-intensive in nature as well. They include Oil & Gas, Defense, Electronic Design Automation (EDA), Life Sciences, Media & Entertainment Studios (M&E), to name just a few. Their needs are illustrated in the following figure.

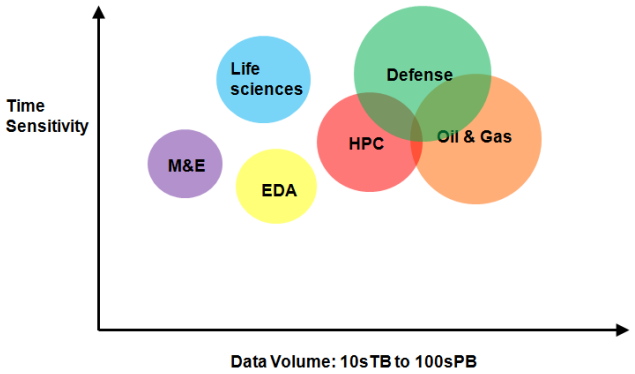

Fig. 1. Six sample data-intensive industries. They all face the challenge of moving massive amounts of data at high-speed. Up to now HPC consists of mostly data-intensive science.

[20] See http://home.cern/about
[21] See http://www.ncsa.illinois.edu/

In the Oil & Gas industry, an in-field exploration operation may need to transfer close to a Petabytes worth of data to remote data centers for in-depth analysis, often over a physical distance of a thousand miles or more[22].

As a consequence, distributed data intensive engineering and science demand a 4th IT dimension, data movement, which actually requires careful and balanced considerations of the other three: storage IOPS, compute power, and network bandwidth - all possibly distributed and/or aggregated from different sources. In addition, the data-growth trends are exponential, thus neither incremental nor one time fixed multiple improvement (e.g. ten times) is sufficient. The employed data transfer solution must scale to match the data growth rate in the coming decade. It must perform well on the existing data management infrastructure. As new hardware and faster infrastructure are introduced, the solution must be able to discover and utilize the newly introduced resources efficiently without time-consuming, interruptive, and tedious upgrades. Furthermore, the solution must optimize the use of space, be energy efficient, cost-effective, and enable the use of modular, commercial off-the-shelf (COTS) [23] hardware. Finally, it must be easy to deploy, manage, and use.

C. *Related observations and remarks*

It is of interest to note that even in 2016, some data intensive industries are still employing data moving approaches involving the shipping of physical media, as illustrated below.

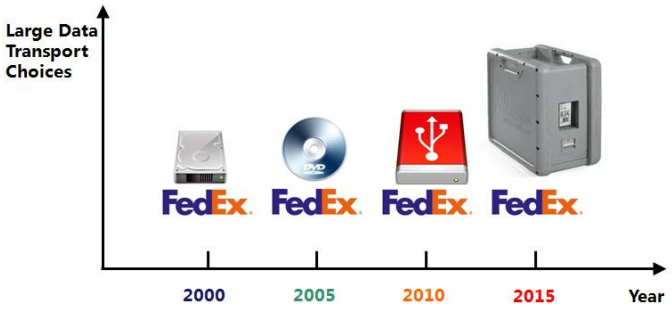

Fig. 2. Shipping physical medium has still been an often employed approach for moving massive amounts of data

On the other hand, network data transfer based approaches are often hampered by data transfer solutions that have not kept pace with the exponential data growth. Many data-intensive enterprises are consequently facing a tsunami of data overrunning transport capability. As a result, there is a considerable loss of productivity and revenue as organizations wait for the data transfers to complete, costing time and therefore money.

[22] See http://webobjects.cdw.com/webobjects/media/pdf/Solutions/Energy-Utilities/High-Performance-Computing-in-Oil-Gas.pdf
[23] See https://en.wikipedia.org/wiki/Commercial_off-the-shelf

It's also of interest to note that high-speed networks are very expensive, as of this writing, $40K/month for a 10G and $400K/month for a 100G. The industry average utilization of such networks is only about 30% - reflecting a very poor infrastructure Return On Investment (ROI).

As more and more enterprises are becoming data-intensive in nature, many are exploring HPC solutions and approaches[24]. Thus creating a high-performance multi-dimensional scalable data transfer solution is critical in keeping and boosting the nation's economic health.

Note that the paper focuses on the data movement (over wide-area networks) challenges faced by highly distributed data-intensive engineering and sciences. It does not focus only on the data movement confined within a "highly optimized commercial datacenter". In addition, with the observed under-utilization of highly expensive networks, our aim is to create a data transfer solution that can achieve high-utilization "*when needed and appropriate*", not "*monopolize all the bandwidth all the time*".

IV. DATA TRANSFER SOFTWARE

A. Areas to improve

Based on our careful review and firsthand experience with the current data transfer solutions, we deem the following areas as the major focus for improvements:

- Lack of scalability and performance from all the commercial tools and most free tools
- Inability to handle data sets of different types equally well from all the tools
- Difficulty in usage to end users for most tools

They are (together with many others) what Zettar ZX is designed for in order to bring huge improvements.

B. Innovations realized

Zettar applies parallel computing in a novel way to scalable data transfers for data-intensive engineering and science. The design goal is to create a data transfer solution that will scale to meet the data transfer requirements for the coming decade, so as to facilitate distributed Exascale computing.

ZX, the software, has a cluster-oriented architecture, using peer-to-peer technologies to ease deployment, operation, usage, and resource discovery. It implements two U.S. patent-pending algorithms to tackle data sets containing lots of small files and very large files, and provide insensitivity to network latency.

Conventional thinking[3] tends to view TCP's sensitivity to loss at high latency as the biggest hindrance to high data transfer speeds. With the two aforementioned algorithms, ZX couples a two-stage data pre-processing and numerous short-duration streams to achieve latency-and-network-error insensitive TCP-based data transfers. Furthermore, the two algorithms enable the application to be deployed on servers with good performance without extensive tuning - a factor that also enables much faster time-to-solution for the product.

With its cluster-oriented architecture and efficient implementation, ZX is capable of providing TLS-encrypted data transfer with a roughly 20% reduction of performance compared to unencrypted data, while using very inexpensive and energy thrifty CPUs such as the Intel E5-2620v3 2.4Ghz[25], without the employment of expensive and purposely built encryption acceleration hardware, e.g. Cavium Nitrox V[26].

In addition, ZX implements unique optimizations enabling effective use of flash storage. Since ZX itself is a cluster and peer-to-peer application, it has the ability to manage and to use raw flash storage devices distributed across the cluster nodes directly. Thus it takes full advantage of such devices' performance potential, without paying the overhead of a local file system and, possibly, the overhead of a parallel file system for aggregating distributed flash devices.

Because of such a capability, if a "fan-out" data distribution pattern is used, a cluster running ZX can make a slow enterprise storage pool look like high-performance HPC storage. ZX can aggregate all flash storage devices across all nodes into a scale-out data transfer buffer. ZX then loads slowly only once when such data is read from the slow enterprise storage pool. All subsequent transfers are done with superior speed.

ZX consists of both a data transfer agent and a management layer - the software has an embedded HTTP server that provides RESTful APIs and a built-in Web User Interface (UI). This removes the complexity and dependency on an external third party service such as used by Globus Online, while keeping the deployment, operation, and usage as simple as possible.

Thus, the management and use of the software, even in a cluster form, can be done via a browser. Furthermore, the RESTful APIs makes it relatively easy to integrate ZX into any workflow; e.g. via command line interface (CLI) scripting - a flexibility that is very important to many users.

All such characteristics not only provide high-performance, but also make it friendly to users and administrators as well.

[24] See http://www.enterprisetech.com/2015/11/19/new-enterprise-users-pushing-hpc-server-growth/

[25] See http://ark.intel.com/products/83352/Intel-Xeon-Processor-E5-2620-v3-15M-Cache-2_40-GHz

[26] See http://cavium.com/processor_security_NITROX-V.html

The high-speed WAN testing set up is also unique and innovative. It is:

1. **Simple**. No complex wiring and very few hardware pieces (most are in place already) are needed
2. **Secure**. By using two VLANs and private IP addresses, the setup isolates itself at SLAC and from the outside
3. **Flexible**. All important network parameters (e.g. latency) can be readily adjusted via coordinating with ESNet
4. **Convenient**. All test servers can stay in the same rack - data travels, not people and hardware - highly cost and time efficient in logistics
5. **Easy to use**. The setup can be switched from LAN to WAN and vice versa with just a couple of commands
6. **Easy to monitor**. ESNet Traffic Map [27] and other monitoring tools all can be used
7. **Versatile and cost-effective.** It leverages the versatile ESNet Test Circuit Service[28]

V. TESTING, EVALUATION, PERFORMANCE, EXPERIMENTAL STUDIES

A. Performance measuring applications

The data transfer performance is measured using several approaches:

1. The software itself has performance counters and statistics output, which are available for review via a Web UI and the CLI.
2. Each cluster node is equipped with a popular run-time statistics software tool, collectd[29] which collects run-time statistics, to be consolidated by another free software tool, graphite [30], and finally displayed with a Grafana dashboard[31].
3. We also use ESNet's Traffic Map extensively.

Note that the 70Gbps file-to-file speed mentioned previously was constrained by the available storage IOPS from a pool formed using eight Intel DC P3700 NVMe Solid State Disks (SSDs) [32], two per node (four nodes per cluster), aggregated using a parallel file system, BeeGFS[33], running on top of XFS[34]. Thus, there is a heavy performance penalty from the two layers of file systems. We anticipate that an external HPC storage tier (most likely all-flash based) will improve the transfer speed by 20% or more. See also **VIII FUTURE**.

[27] See https://my.es.net/
[28] See http://www.es.net/network-r-and-d/experimental-network-testbeds/test-circuit-service/
[29] See https://collectd.org/
[30] See https://github.com/graphite-project/
[31] See http://grafana.org/
[32] See http://www.intel.com/content/www/us/en/solid-state-drives/ssd-dc-p3700-spec.html
[33] See http://www.beegfs.com/content/
[34] See http://xfs.org/index.php/Main_Page

B. System and environments

Two different environments are used:

a) Actual production DTNs

We have been using an existing DTN employed by the LCLS project on the SLAC campus, and another DTN operated by NERSC in Berkeley, California (about 2.5ms RTT from SLAC) for performance testing. The ZX software is installed on both DTNs. The two ZX instances are remotely controlled via their respective Web UI. We have used various data sets of 200GiB each, consisting of files sized to the commonly observed sizes by LCLS Data Management. The bbcp application has been used to generate baseline data, supplemented with test results using Globus and GridFTP. All tests have been conducted so far at various times of the day and days of a week, so as to make the results as realistic as possible. The following are the main hardware specifications for the LCLS and NERSC DTNs. Note the hardware age, capability differences, and very critically, the performance (or the lack thereof) of the storage pool used on each end.

	LCLS	**NERSC**
OS	RHEL 6.x and 7.1	RHEL 6.7
CPU	Dual Xeon(R) CPU E5520 @ 2.27GHz (*launched in Q1 '09*)	Dual Xeon(R) CPU E5-2680 v2 @ 2.80GHz (*launched in Q3'13*)
Memory	24GiB	128GiB
Network storage	Lustre pool (*not striped*)	IBM GPFS pool (*striped*)
Ethernet	10GbE for connecting to ESNET (100GbE)	4x10GbE (bonded) for connecting to ESNET (100GbE)
Infiniband	QDR connection to the LCLS Lustre file systems	FDR connection to the LCLS Lustre file systems

b) PoC set up

We employ two 2 Rack Unit (RU) 4 Node high-density servers [35] provided by Quanta Cloud Technology [36] so as to form two four-node clusters. Each node of the two servers is equipped with 4x10G + 1x40G Ethernet ports, dual Intel E5-2620v3 CPUs, 128GiB RAM, one Intel DC S3500 SSD[37] as the system drive, and another DC S3500 as the BeeGFS' metadata drive, together with two Intel DC P3700 NVMe SSDs as built-in data storage.

Each server is connected to an Arista 7280SE-68 10/100G switch[38] and a Quanta T3040 LY08 10/40G switch[39]. One of

[35] See http://www.qct.io/Product/Server/Rackmount-Server/Multi-node-Server/QuantaPlex-T41SP-2U-4-Node-p283c77c70c83c185
[36] See http://qct.io/
[37] See http://www.intel.com/content/www/us/en/solid-state-drives/ssd-dc-s3500-spec.html
[38] See https://www.arista.com/en/products/7280e-series
[39] See http://www.qct.io/Product/Networking/Ethernet-

the Arista 7280SE-68 switches is connected to the SLAC 100Gbps border router as shown in Fig. 3. Fig. 4 illustrates the per cluster setup.

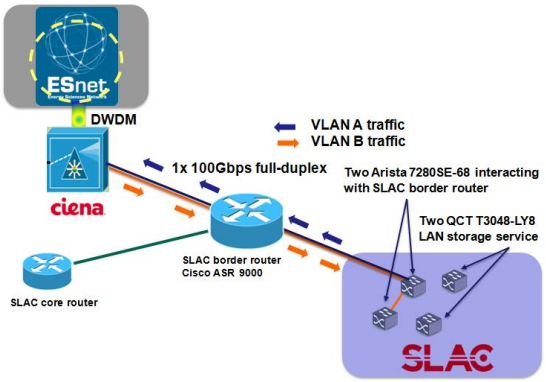

Fig. 3. The WAN Testing high-level topology

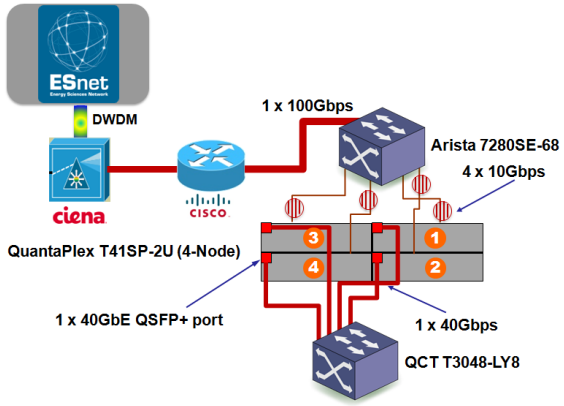

Fig. 4. Simplified WAN testing diagram for each cluster

Each node also runs as both a BeeGFS storage server and storage client. The reason why we run BeeGFS this way is to keep the entire PoC setup compact, self-contained, and mobile. More info can be found in the SLAC-TN-15-001[4].

Furthermore, since the Arista 7280SE-68 switch has two 100G ports, they can be employed to form a 200G link-aggregation[40] channel for testing ZX data transfer performance at greater than 100G in a LAN environment. This is how the to-be-discussed 155Gbps and 120Gbps memory-to-memory results were obtained. The clean and flexible test setup makes this very easy to do.

VI. Performance Results

A. Results from the production systems

From mid-March to early April, using two existing DTNs at SLAC and NERSC, extensive comparison testing was conducted comparing bbcp and ZX's data transfer performance with normal test data sets, 200GiB each, at different times of the day and different days of the week. From below it's evident that on a per-host basis, the two are comparable.

Software	Range[41]
bbcp	174.5 MB/s - 808.2 MB/s (1.36 Gbps - 6.31 Gbps)
ZX	352 MB/s - 640 MB/s (2.75 Gbps - 5.0 Gbps)

Specifically for ZX, two typical outcomes are tabulated below:

	4x50GiB			10x20GiB		
Test #	**Run time (secs)**	**Transfer speed (Gbps)**	**Peak speed (Gbps)**	**Run time (secs)**	**Transfer speed (Gbps)**	**Peak speed (Gbps)**
1	364	4.72	6.87	375	4.58	8.05
2	332	5.17	8.00	360	7.17	8.16
3	257	6.68	7.78	228	7.53	8.43
Avg	318	5.52	7.55	321	6.43	8.21
Median	332	5.17	7.78	360	7.17	8.16
Std. Dev.	55	1.03	0.60	81	1.61	0.20

B. Results from the PoC setup

Using the PoC setup we measured the impact of applying TLS-encryption on the transfer speed. The impact as shown below is a roughly 20% reduction in transfer speed.

a) 155Gbps (unencrypted) memory-to-memory over a 2x100Gbps aggregated links and 16x10Gbps network interfaces

Test #	Transfer speed (Gbps)	Peak Speed (Gbps)
1	158.031	158.560
2	157.660	158.404
3	156.845	157.706
Average	157.512	158.223

b) 120Gbps (TLS-encrypted) memory-to-memory over a 2x100Gbps aggregated links and 16x10Gbps network interfaces

Test #	Transfer speed (Gbps)	Peak speed (Gbps)
1	122.552	124.494
2	122.474	124.252
3	123.765	125.557
Average	122.930	124.767

Switch/T3000-Series/QuantaMesh-T3048-LY8-p43c77c75c158c205

[40] See https://en.wikipedia.org/wiki/Link_aggregation

[41] The ranges quoted here are the minimum and maximum average throughput from more than thirty runs; the typical run time, depending on the speed, is around 10 – 15 minutes.

C. File-to-file for data sets containing large files

We present the results using a test data set consisting of 20000x50MiB files; ~ 1TiB in overall size. Assuming each employed test data set consists of files of the same size, e.g. 20000x50MiB, 2000x500MiB, 400x5GiB etc., they show transfer speeds of over 70Gbps. The one shown below, 20000x50MiB, has an average speed in the middle of the pack. All test data sets were transferred over the 5000 mile 100Gbps link, with TLS-encryption.

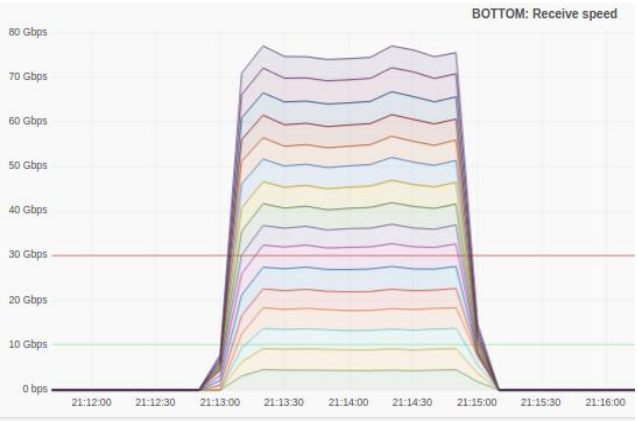

Fig. 5. Grafana site-to-site dashboard shows the data transfer speed profile time histories for the receiving side. Test data set employed consists of 20000x50MiB files. Note the even utilization of all 16 network interfaces. It also of interest to note that no interface-bonding is needed

D. File-to-file for data sets containing lots of small files

We also present the result using a LOSF test data set consisting of 1000000x1MiB; ~1TiB in size, transferred with TLS-encryption over a 5000 mile 100Gbps link. The transfer speed is about 30Gbps.

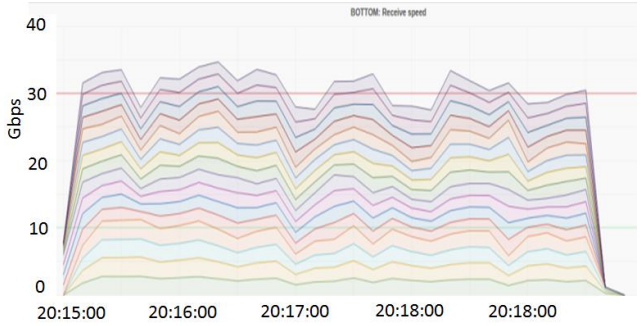

Fig. 6. Grafana site-to-site dashboard shows the data transfer speed profile time histories for the receiving side. Test data set employed consists of 1000000x1MiB files. Note again the even utilization of all 16 network interfaces

VII. CONCLUSIONS/DISCUSSIONS

Unlike most software, *it's extremely costly to create a production-ready, next generation high-performance, multi-dimensional scalable data transfer software solution.* For critical tasks such as validation, testing, software behavior analysis, and profiling, the developers *must have access to* established facilities and an operational high-speed network infrastructure worth hundreds of millions of dollars, as well as more than a million dollars worth of leading edge hardware (100Gps switches, high-end servers, NVMe SSDs, HCAs[42], HBAs[43], optical cables and transceivers, and one or more highly costly 100G multi-port router add-on cards).

Most open source software developers (or even commercial software vendors) do not have access to the necessary resources to do the job right. It must be noted that *mission-critical data management software must be validated via real-world testing. Lab simulations alone cannot prove usability* - software for high-speed, massive data transfers falls into this category. Next let's review and comment on the data movement challenges that data-intensive engineering and science face today and in the future.

As a group, we know of the stringent data management requirements, including data transfer rates, of top-tier science projects well. Few of the current solutions, commercial and free, including government funded entries, can meet all the performance, scalability, cost, work-flow integration, and usability requirements that we deem valuable. We also know that to meet these data management demands, an effective solution takes time to create – a mission-critical data management infrastructure software often demands a multi-year effort in development, validation, and testing, especially the last two. As a result, we must act early, learn early, test early, test often, test realistically in a production environment and on a real infrastructure, and get the solution ready before the actual demands arise – we can't afford fire-fighting. Our test data sets are often of TiB sizes of various file sizes and types - some of them contain millions of file entries. Within this past year, we have test transferred PiBs worth of data. We want to cover a testing envelope that most users won't even exceed for a long time. The Principle of Least Surprise[44] is a critical key in the creation of such software.

Thus, this paper has devoted much space describing how we have been rigorously and methodologically testing the ZX software so far. We have planned to carry out more testing using ESnet and possibly research networks in other parts of the world in the future. This is to ensure that the software, once deployed, can really meet the production demands.

Earlier in this paper, we also mentioned RESTful API support for scripting and work-flow integration. This is a very important aspect to large scale data-intensive engineering and science projects. Thus, a short discussion should be in order: based on what we have seen, the commercial solutions tend to emphasize the end-user facing aspects, for example, graphical

---

[42] See http://www.webopedia.com/TERM/H/Host_Channel_Adapter.html

[43] See https://en.wikipedia.org/wiki/Host_adapter

[44] See https://en.wikipedia.org/wiki/Principle_of_least_astonishment

user interfaces (GUIs) and host-based mechanisms. For serious data-intensive engineering and science projects, the demand for workflow integration is always the ability to enable hands-off automation and remote control, as opposed to user-facing and interactive operations. Thus, how well a solution can programmatically support such a demand, in a scale-out manner is of paramount importance.

Another aspect is that modern data-intensive engineering and science projects are often highly distributed. We have given some examples already. There are many more.

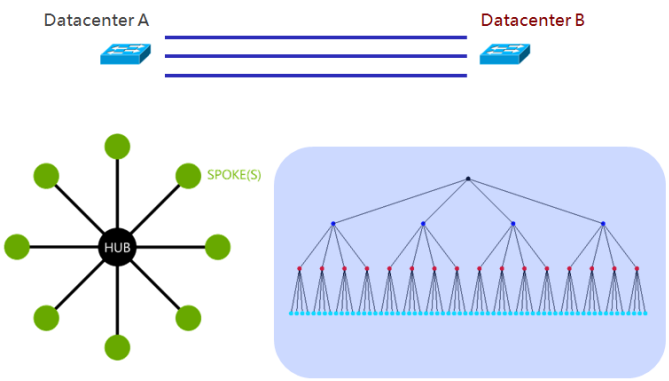

Fig. 7. Data distribution patterns common to data-intensive engineering and science projects

As a consequence, how well a solution behaves in handling the data distribution patterns involved in such projects is also of critical importance to the efficiency of the execution and eventually the success of each project. By design, Zettar ZX handles the three patterns shown in Fig. 7.

Nevertheless, "by-design" alone is not good enough for tier-1 data-intensive projects. We must rigorously carry out realistic tests to actually validate a solution's capability to ensure it would meet the production requirements when time comes. For mission critical data management infrastructure software, this should be deemed as part of the solution creation process. No one can afford to take a short-cut. We will discuss more what we plan to do in the next section.

To summarize, this paper's main contributions are in the following aspects:

1. Our findings and results shed a much needed light to the members of the data-intensive communities (both commercial and research) regarding the importance of treating data movement as the 4th critical IT dimension, besides the traditional three: storage, compute, and networking, together with the ever increasing importance of multi-dimensional scalability.
2. We showed that the 4th-dimension must be defined by the balanced interactions of the first three to achieve a desired data transfer speed, e.g. 400Gbps.
3. We have shown that the use of standard TCP can meet the performance needs without the need of a specialized protocol. This enables us to tap into the continual and numerous improvements from both commercial entities[45] and the open-source community[46].
4. We reported our experience using a new reference data transfer system design, as described in SLAC Technical Note: SLAC-TN-15-001[4]. Comparing to the existing state-of-the-art[47], it is more scalable, space and energy efficient, cost-effective, and easy to deploy and use. It is likely to inspire a new generation of high-performance DTNs and appliance designs.
5. In addition, we have demonstrated that we have met the data transfer needs of major scientific experiments for this decade. This should foster accelerated progress in science and engineering, thus the overall national economic health.

VIII. FUTURE

We have made extensive comparisons between Zettar ZX and bbcp, plus preliminary comparisons with GridFTP. We deem it necessary to carry out more extensive comparisons with GridFTP. We also plan to include comparisons with XRootD and where feasible commercial solutions, as part of future work. This will include using the cluster setup for all, so as to gain a fact-based and balanced view of the strength and weakness of each solution. We believe that only in this way, we will be in a position to recommend an appropriate solution to users involved in data-intensive engineering and science endeavors.

In the near term we expect to make memory to memory transfers between multiple DTNs at one site to multiple DTNs at the second site, all running ZX. This will demonstrate the handling of complex distribution patterns. In addition we plan to make file to file transfers from a PoC cluster with a parallel file system layered on top of a pool of SSDs, to multiple independent DTNs front ending a parallel file system. The goal is to demonstrate we can mix and match existing independent DTN setups running ZX with a cluster of collaborative DTNs. This in turn enables a practical and scalable migration path.

Furthermore, given the critical need for high storage IOPS for high-speed massive data transfers, it's almost mandatory to use a really high-performance HPC storage tier as part of a data transfer setup. The state of the art, confronting the multi-100Gbps demand that data-intensive science and engineering will soon face in this and the coming decade, simply is

[45] See e.g. http://googlecode.blogspot.com/2012/01/lets-make-tcp-faster.html

[46] See e.g. https://lwn.net/Articles/629155/

[47] See https://fasterdata.es.net/science-dmz/DTN/reference-implementation/

inadequate in this aspect[48]. Fig. 8 should make the present situation clear.

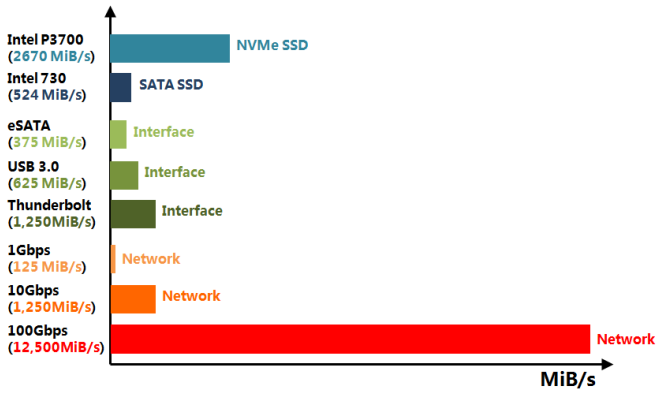

Fig. 8. Comparing the speed of Intel DC P3700 NVMe sequential R/W (quoted from its spec), ditto for Intel 730, eSATA, USB 3.0, Thunderbolt, and three common network speeds, including the current state of the art 100Gbps, it should be evident that the throughputs of common storage devices, even the advanced NVMe SSDs, pale in comparison with the speed of a 100Gbps network.

Fortunately, with the rapid advances of flash technologies; e.g. high-performance and high-capacity NVMe SSDs such as Intel's DC P3700 and DC P3320[49] families, compact, high-performance 1 or 2U servers such as QCT's QuantaGrid D51BP[50] and AIC's SB122A-PH[51], the ready available high-performance parallel file systems such as Intel Lustre[52] or ThinkParQ [53] 's BeeGFS, together with high-speed and efficient Ethernet and Infiniband interconnects from vendors such as Arista[54] and Mellanox[55], it's quite feasible to create an all-flash and scale-out capable HPC storage tier to support the high-speed data transfers with low-latency and low-overhead.

While on the subject of using an all-flash HPC storage tier, it should be appropriate to discuss the available interconnect and kernel-offload/bypass choices. Remote Direct Memory Access (RDMA)[56] is a technology that allows data to be written from one computer directly into the memory of another computer over a network connection. This bypasses many of the operating systems and network stacks that slow down transfers. RDMA has long been associated with InfiniBand, and is, in the views of many[57], the only reason for the existence of that link type. In our first hand experience, and that of many of our colleagues', InfiniBand deployments often do see higher performance and lower latencies - that the combination of InfiniBand and RDMA is widely used in the high-end HPC centers has well-grounded justifications. Nevertheless, we also know of the simplicity and popularity associated with Ethernet, which also has RDMA capability, via RoCE[58] or iWRAP[59]. In addition, there are emerging competitions to Infiniband, such as Intel's OmniPath[60]. All such options should be reviewed and tested, again to gain a firmly grounded understanding, which in turn should enable us to make the right choice for any deployment scenario.

In passing, it should be of interest to note that RDMA can be regarded as a kind of kernel bypass technology[61], so it ought to be fruitful to compare its benefits vs that of other recent, software-based approaches, as represented by Intel's DPDK[62].

Finally, we will also note that with some of the top-tier data-intensive science and engineering projects, the data sources could be many thousands or more sensors which continuously emit data streams at very high-speed. Due to the local computing power availability (or the lack thereof), the many thousands of fast growing files thus generated must be transferred to a location with sufficient computing power available for effective analysis. It should be evident that nearly real-time data transfers for such growing files are a mandatory part of the overall workflow. This type of demand far exceeds the capabilities of what have been available from free solutions, based on e.g. ionotify [63] or the "hot-folder" mechanisms[64] often seen in commercial solutions used by e.g. the media and entertainment industry. This is an area that the authors of this paper will tackle as a future effort as well.

We gratefully acknowledge the support of SLAC, in particular that of John Weisskopf and Ron Barrett for installing the equipment, Teri Church for assistance with loaning the equipment, Antonio Ceseracciu for network expertise, Yemi Adesanya and Amedeo Perazzo for discussions, positive criticism and challenges, and James Williams, Chris Kennedy

[48] See http://fasterdata.es.net/science-dmz/DTN/reference-implementation/; with 16 250G SATA-3 HDD drives, it can sustain only 2.2GByte/sec. (17.7Gbps) reading from disk. See also Fig. 6.
[49] See http://www.intel.com/content/www/us/en/solid-state-drives/ssd-dc-p3320-brief.html
[50] See http://www.qct.io/Product/Server/Rackmount-Server/1U/QuantaGrid-D51BP-1U-p277c77c70c83c85?search=D51bp
[51] See http://www.aicipc.com/ProductDetail.aspx?ref=SB122A-PH
[52] See http://www.intel.com/content/www/us/en/software/intel-enterprise-edition-for-lustre-software.html
[53] See http://thinkparq.com/
[54] See http://www.arista.com/en/
[55] See http://www.mellanox.com/
[56] See https://en.wikipedia.org/wiki/Remote_direct_memory_access
[57] See e.g. http://www.networkcomputing.com/networking/will-rdma-over-ethernet-eclipse-infiniband/1893518522
[58] See https://en.wikipedia.org/wiki/RDMA_over_Converged_Ethernet
[59] See https://en.wikipedia.org/wiki/IWARP
[60] http://www.intel.com/content/www/us/en/high-performance-computing-fabrics/omni-path-architecture-fabric-overview.html
[61] See http://developers.redhat.com/blog/2015/06/02/can-you-run-intels-data-plane-development-kit-dpdk-in-a-docker-container-yep/
[62] See http://dpdk.org/
[63] See http://man7.org/linux/man-pages/man7/inotify.7.html
[64] See http://filecatalyst.com/solutions/filecatalyst-direct/schedulingand-automation/

and Theresa Bamrick for their continued support. The support of ESnet was critical in providing access to a 100Gbps circuit and routing it via Atlanta. The support and partnership with AIC, Arista, Cisco, Dell, Intel, Mellanox, QCT and ThinkparQ for the loans of equipment and software was invaluable for the PoCs. We also wish to give credit to the innovative and excellent software engineering of Igor Soloviov and Oleksandr Nazarenko, both of Zettar Inc., for the robust Zettar ZX implementation.

REFERENCES

[1] A. Rajendran, P. Mhashilkar, H. Kim, D. Dykstra, G. Garzoglio and I. Raicu, "Optimizing Large Data Transfers over 100Gbps Wide Area Networks," in *13th IEEE/ACM International Symposium on Grid, Cloud and Grid Computing (CCGrid)*, 2013.

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

[3] "Ultra High-speed Transport Technology. IBM/Aspera white paper," [Online]. Available: http://asperasoft.com/resources/white-papers/ultra-high-speed-transport-technology/. [Accessed 30 March 2016].

[4] C. Fang, "Using NVMe Gen3 PCIe SSD Cards in High-density Servers for High-performance Big Data Transfer Over Multiple Network Channels," SLAC Technical Note, 2015.