Measurements and Analytics of Wide-Area File Transfers over Dedicated Connections

Nageswara S. V. Rao
Oak Ridge National Laboratory
Oak Ridge, TN, USA
raons@ornl.gov

Qiang Liu
Oak Ridge National Laboratory
Oak Ridge, TN, USA
liuq1@ornl.gov

Satyabrata Sen
Oak Ridge National Laboratory
Oak Ridge, TN, USA
sens@ornl.gov

Zhengchun Liu
Argonne National Laboratory
Argonne, IL, USA
zhengchun.liu@anl.gov

Raj Kettimuthu
Argonne National Laboratory
Argonne, IL, USA
kettimut@anl.gov

Ian Foster
Argonne National Laboratory
Argonne, IL, USA
foster@anl.gov

ABSTRACT

Distributed scientific and big-data computations are becoming increasingly dependent on access to remote files. Wide-area file transfers are supported by two basic schemes: (i) application-level tools, such as GridFTP, that provide transport services between file systems housed at geographically separated sites, and (ii) file systems mounted over wide-area networks, using mechanisms such as LNet routers that make them transparently available. In both cases, transfer performance depends critically on the configuration of associated host, file, IO, and disk subsystems, each of which is complex by itself, as well as on their complex compositions, implemented using buffers and IO-network data transitions. We present extensive file transfer rate measurements collected over dedicated 10 Gbps connections with 0–366 ms round-trip times, using GridFTP and XDD file transfer tools, and the Lustre file system extended over wide-area networks with LNet routers. Our test configurations are composed of: three types of host systems; XFS, Lustre, and ext3 file systems; and Ethernet and SONET wide-area connections. We present analytics based on the convexity-concavity of throughput profiles which provide insights into throughput and its superior or inferior trend compared to linear interpolations. We propose the utilization-concavity coefficient, a scalar metric that characterizes the overall performance of any file transfer method consisting of specific configuration and scheme. Our results enable performance optimizations by highlighting the significant roles of (i) buffer sizes and parallelism in GridFTP and XDD, and (ii) buffer utilization and credit mechanism in LNet routers.

CCS CONCEPTS

• **Networks** → **Network performance evaluation**; **Network measurement**;

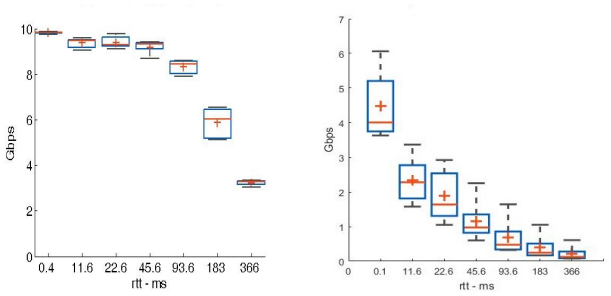
Keywords and phrases: Lustre file system, wide-area networks, throughput profile, network measurements.

ICDCN '19, January 4–7, 2019, Bangalore, India

1 INTRODUCTION

Big data and science applications are becoming increasingly distributed, and they often require coordinated computations at geographically distributed sites that require access to files over Wide-Area Networks (WANs). Typically, file systems are installed at local sites, and wide-area file transfers are carried out by tools, such as Globus [4], Aspera [3], XDD [23], UDT [5], MDTM [10], and others. Another less frequently used scheme enhances the infrastructure to mount file systems over WAN [6, 11], thereby making them transparently available at remote sites. In particular, the Lustre file system, typically implemented over an InfiniBand (IB) site network, may be extended to Ethernet WAN using LNet routers [14]. File transfer throughputs achieved in these scenarios vary significantly based on two factors: (i) *configuration* that consists of file and storage system, transfer hosts and network connections, and (ii) file transfer *scheme*, such as GridFTP or LNet router. We collectively refer to both as the *file transfer method*. In this paper, we consider file transfers that utilize the Transmission Control Protocol (TCP) for underlying data transport over dedicated connections, such as those provisioned by ESnet's OSCARS [12] and Google's Software Defined Network (SDN) [7].

The performance of a file transfer method is characterized by its *throughput profile*, $\hat{\Theta}_E(\tau)$, which specifies throughput achieved over a connection as a function of Round-Trip Time (RTT) τ. A throughput profile critically depends on the configuration, and equally importantly, on compositions of file systems and network connections at the sites, which involve matching file IO with network transport through buffer management. For example, over dedicated 10GigE connections with $\tau \in [0, 366]$ ms, Fig. 1(a) and Fig. 1(b) show $\hat{\Theta}_E(\tau)$ for XDD with an XFS file system mounted on Solid State Device (SSD) storage, and a Lustre file system extended using LNet routers, respectively. We see two contrasting profiles that differ evidently in both peak throughput values (10 and 4.5 Gbps) and shape (*concave* and *convex*). The peak of a throughput profile is a direct indicator of the performance, and furthermore, its shape is subtler but also an important indicator, particularly, in projecting measurements to obtain throughput estimates [9, 15]. Consider projecting $\hat{\Theta}_E(\tau)$ based on throughput measurements collected at RTTs τ_1 and τ_2, for $\tau_1 < \tau < \tau_2$. For concave profiles, $\hat{\Theta}_E(\tau)$ is guaranteed to be above the linear interpolation of $\hat{\Theta}_E(\tau_1)$ and $\hat{\Theta}_E(\tau_2)$, which is a highly desirable property. On the other hand, for convex profiles, the only guarantee that can be provided is that $\Theta_E(\tau)$ is



(a) XDD transfers between XFS file systems mounted on SSDs

(b) Lustre file transfers using LNet routers

Figure 1: Throughput profiles $\hat{\Theta}_E$ of file transfers over 10GigE connections

above the minimum of $\hat{\Theta}_E(\tau_1)$ and $\hat{\Theta}_E(\tau_2)$. At a deeper level, the shape of $\hat{\Theta}_E(\tau)$ reflects the time dynamics of file transfers, and in particular its concavity requires fast ramp-up followed by stable throughput [15].

Overall, the current and past measurements indicate that $\hat{\Theta}_E(\tau)$ is typically concave for smaller RTT and then switches to convex as RTT is increased [15–18]. Consider the memory transfer profile $\hat{\Theta}_T(\tau)$ of the underlying TCP transport, which itself exhibits such a dual property, typically with a wider concave region [18]. High-performance protocols, such as Hamilton TCP [21] and Scalable TCP [8], provide wider concave regions compared to other TCP variants. Furthermore, increasing configuration parameters, such as the buffer size and number of parallel streams, leads to expanded concave regions [15], which represents a more effective transport method. The file transfer profile $\hat{\Theta}_E(\tau)$ is a modulated version of $\hat{\Theta}_T(\tau)$ by file, IO and host systems, typically resulting in smaller concave regions, as shown subsequently. In this paper, we show the dependencies of $\hat{\Theta}_E(\tau)$ on both network and file IO parameters, and highlight the differences among various file transfer methods, and then propose systematic methods to compare them. We mainly focus on analytics of measurements under various configurations that have described in detail in [14–17, 19], which we briefly summarize here in various sections to provide the context and illustrate specific performance parameters.

We summarize various file throughput measurements collected over a suite of dedicated 10 Gbps emulated connections with 0-366 ms round-trip times, using GridFTP [22] and XDD [23] file transfer tools, and Lustre file system extended over WANs using LNet routers [14]. Their collection spans a five-year period and covers a variety of combinations with large Terabyte datasets, and the individual configurations and tests have been described in detail in [14–17, 19]. Together, these test configurations are composed of: (i) three types of host systems, 48- and 32-core data transfer servers, and 32-core cluster nodes; (ii) three file systems, host file systems ext3 on local hard disk, XFS on SSD drive, and Lustre implemented on IB network; and (iii) two types of network connections, 10 Gbps Ethernet, and 9.6 Gbps OC192 SONET wide-area connections. As expected, in reflecting TCP transport, $\hat{\Theta}_E(\tau)$ decreases with RTT and has narrower concave regions that reflect the file transfer methods, but their comparison is complicated by different peaks and convex-concave transitions.

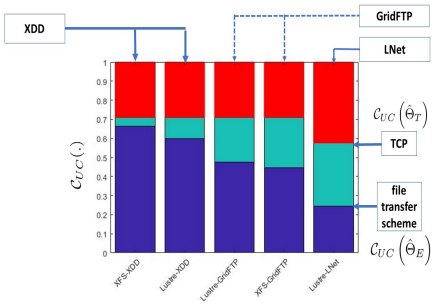


Figure 2: Summary of C_{UC} for five file transfer methods

To provide insights into throughput profile and its trends, we present analytics based on the convexity-concavity geometry of $\hat{\Theta}_E(\tau)$ using the above collection of measurements as a starting point. Specifically, we propose the *utilization-concavity coefficient* $C_{UC} \in [0, 1]$, a scalar metric that represents both peak throughput and concave region of $\hat{\Theta}_E(\tau)$. It is as an extension of a simpler memory transfer version in [9], and it enables an objective comparison of multiple file transfer methods. For five different file transfer methods[1], C_{UC} is illustrated in Fig. 2 in blue, wherein left- and right-most columns represent the best and worst profiles corresponding to XDD XFS and Lustre LNet transfers in Figs. 1(a) and 1(b), respectively. The green regions represent additional throughput achieved by TCP memory transfers, and the red regions represent the connection capacity not utilized by TCP. In addition to performance comparison, our results show that these analytics lead to performance optimizations by highlighting the roles of parallelism in GridFTP and XDD, and buffer utilization and credit mechanism in LNet routers.

The organization of this paper is as follows. File systems and transfer tools are briefly described in Section 2. In Section 3, we introduce the utilization and concave-convex coefficients to characterize the file transfer throughput profiles. The testbed utilized in collecting the measurements is discussed in Section 4. Throughput measurements and profiles are presented in Section 5, and their analytics are described in Section 6. Statistical guarantee aspects of the utilization-concavity coefficient are discussed in Section 7. We conclude in Section 8.

2 FILE SYSTEMS AND TRANSFER TOOLS

A wide-area disk-to-disk file transfer configuration encompasses storage devices, data transfer hosts, and local- and wide-area connections, as illustrated in Fig. 3. Several sites often use dedicated data transfer hosts, such as Data Transfer Nodes (DTNs), along with the high-performance Network Interface Cards (NICs) to access network connections and the Host Channel Adapters (HCAs) to access network storage systems. Transfers also involve a range of software components, including the file system IO modules for disk access and the TCP/IP stack for network transport. The file throughput profile $\hat{\Theta}_E(\cdot)$ is composed of the underlying TCP profile $\hat{\Theta}_T(\cdot)$ and the local file throughput $\hat{\Theta}_H(\cdot)$ at the host H. Typically, $\hat{\Theta}_E(\tau) \leq \hat{\Theta}_T(\tau)$ and they both decrease with τ, and $\hat{\Theta}_H$ at end

[1]We did not attempt to extensively optimize the individual methods, since our goal for this figure is to illustrate and highlight different profiles.

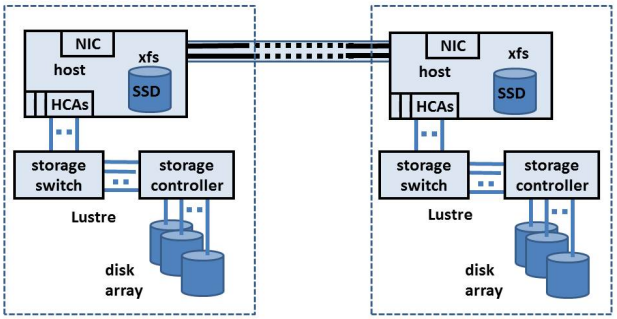


Figure 3: File transfers over long-haul connections

nodes limits the peak throughput of $\hat{\Theta}_E(\tau)$, which is achieved at lower RTT τ. The resultant performance critically depends on the parameters of file I/O and network transport, which are parts of the file transport method. In particular, parallel operations are used to increase the effective throughput of both file systems and wide-area network transport, typically by utilizing a set of buffers. We will now briefly describe the coordination of these two critical systems by XDD, GridFTP, and LNet routers.

2.1 XDD File Transfers

For file transfers, XDD spawns a set of threads to open a file and perform data transfers between storage and network. A source XDD process creates a *TargetThread* that opens the file, initiates a connection with a destination XDD process, and subsequently creates a number of *QThreads* that issue read commands to fill a thread-local buffer. Once a thread's buffer is filled, that thread transmits the data over the network to a destination XDD process. Similarly, the destination XDD process initiates a thread that listens for a connection from a source XDD process and then creates QThreads to receive data from the network and write the data into the storage system. The number of source and destination QThread pairs is equal to the number of TCP parallel streams. XDD reports *read* transfer rate at the sender and *write* transfer rate at the receiver for each file transfer by aggregating across all flows. This tight pairing of IO and TCP streams is particularly effective in Lustre file systems when the number of flows matches the number of stripes used for file operations [16].

2.2 GridFTP File Transfers

GridFTP is an extension of the standard File Transfer Protocol (FTP) for high-speed, reliable, and secure data transfer [1]. It implements extensions to FTP, which provide support for striped transfers from multiple data sources. Data may be striped or interleaved across multiple servers, as in a parallel file system such as Lustre. GridFTP supports parallel TCP flows via FTP command extensions and data channel extensions. A GridFTP implementation can use long virtual round trip times to achieve fairness when using parallelism or striping. In general, GridFTP uses striping and parallelism in tandem, i.e., multiple TCP streams may be open between each of the multiple servers participating in a striped transfer. However, this process is somewhat different compared to XDD wherein each IO stream is handled by a single TCP stream, whereas such association is less strict in GridFTP.

2.3 Lustre over Wide-Area Networks

Lustre file system employs multiple Object Server Targets (OSTs) to manage collections of disks, and multiple Object Storage Servers (OSSs) to stripe file contents. Lustre clients and servers connect over the network, and are configured to match the underlying network modality, for example IB or Ethernet. Host systems are connected to IB switch via HCAs, and Lustre over IB clients is used to mount the file system on them over IB connections. Due to a latency limit of 2.5 ms, such deployments are limited to site-level access, and do not provide file access over wide-area networks. This IB-based Lustre file system is augmented with Ethernet Lustre clients, and LNet routers are utilized to make IB-based OSSs available over wide-area Ethernet connections [14]. Compared to GridFTP and XDD that are software applications, the implementation of LNet routers requires more changes to the infrastructure.

2.4 Host and TCP Profiles

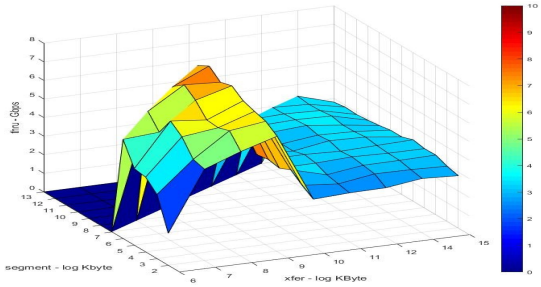
We now consider cases that illustrate the parameters that affect host profiles $\hat{\Theta}_H$ and TCP profiles $\hat{\Theta}_T$, which in turn affect $\hat{\Theta}_E$.

2.4.1 Host Profiles. To establish baselines for $\hat{\Theta}_H$ we consider IOzone measurements collected for four host configurations:

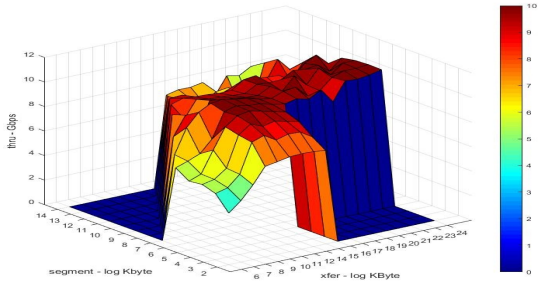
(i) `feynman`: 32-core workstation with ext3 file systems mounted on local hard disks,
(ii) `bohr`: 48-core server with XFS file system mounted on SSD drives connected over PCI bus,
(iii) `bohr`: 48-core server with Lustre mounted on IB and Ethernet using LNet routers, and
(iv) `tait`: 32-core cluster node with Lustre mounted on IB and Ethernet using LNet routers.

IOzone disk write measurements are shown in Fig. 4 corresponding to different transfer sizes and segment sizes for cases (i) and (ii). For ext3 in case (i), the throughput is a function of the transfer size as shown in Fig. 4(a), wherein throughput rates are much higher for smaller sizes but are limited to 2 Gbps for 10 GB transfers that access the disk system. For XFS on SSD, the write throughput rates are about 10 Gbps, and are maintained for 10 GB data transfers as shown in Fig. 4(b), thereby illustrating the higher speed of SSD devices.

For Lustre under IB and Ethernet configurations, IOzone throughput measurements are shown in Fig. 5. The peak write throughput $\hat{\Theta}_H$ of 6 Gbps is achieved in both cases in transferring datasets of size 1 GB by using a sufficiently large segment size. However, $\hat{\Theta}_H$ is maintained around peak 6 Gbps in IB configuration and significantly decreased for LNet configuration for 2 GB (or larger) data transfers. In particular for 10 GB transfers, $\hat{\Theta}_H$ in the latter case decreases to around 2 Gbps, whereas it is maintained at 6 Gbps in IB configuration, and increasing the number of stripes from 2 to 8 has a limited effect. These LNet router effects are critical in determining not only the peak but also the convexity of $\hat{\Theta}_E(\tau)$ as will be discussed subsequently. Thus, IOzone measurements highlight the critical differences between $\hat{\Theta}_H$ under different file systems and storage media.

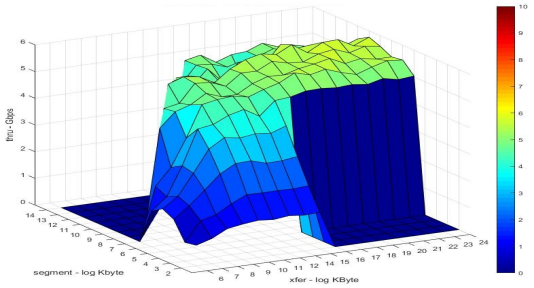


(a) ext3 mounted on local hard disk

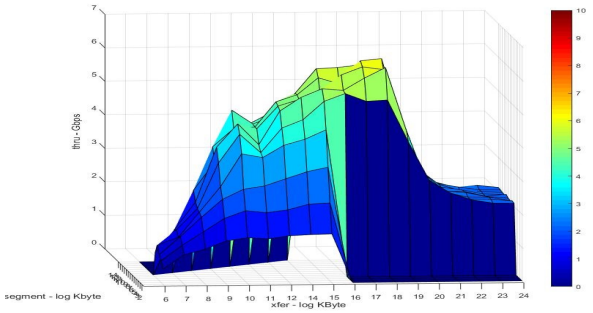


(b) xfs mounted on SSD

Figure 4: IOzone write throughput measurements: ext3/xfs

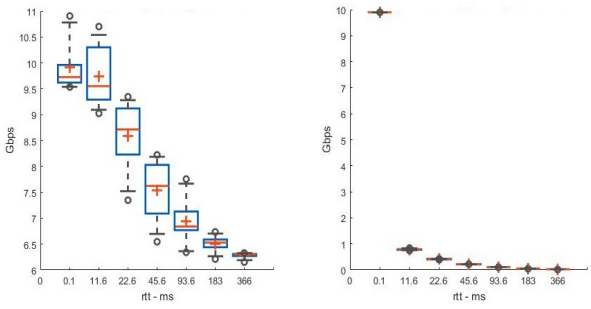


(a) Lustre IB clients supported by site IB network



(b) Lustre Ethernet clients supported by Lnet routers

Figure 5: IOzone write throughput measurements: Lustre



(a) 48-core data transfer hosts with tuned TCP parameters

(b) 32-core cluster nodes with default TCP parameters

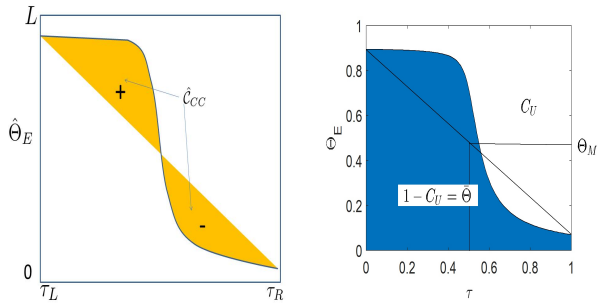
Figure 6: Throughput profiles of TCP memory transfers for data transfer and cluster nodes

2.4.2 TCP Profiles. We obtain $\hat{\Theta}_T$ for two classes of hosts (a) 48-core and 32-core stand-alone systems (`bohr` and `feynman`, respectively), which are typically used for data transfers, and (b) 32-core hosts (`tait`) which are nodes of a compute cluster typically used for computations. The hosts are configured to use Hamilton TCP (HTCP) [21] in most of our tests, and their buffer sizes are set at largest allowable values. TCP parameters of data transfer hosts are tuned for wide-area connections, which achieve peak memory transfer rates measured by iperf shown in Fig. 6(a). On the other hand, cluster nodes have default parameters that achieve much lower memory transfer rates shown in Fig. 6(b). To support LNet connections, their TCP buffers are changed to the larger values to avoid TCP being the throughput bottleneck. The resultant shape of $\hat{\Theta}_T$ reveals critical performance parameters in addition to improved throughput. The improved profiles are concave for lower RTT and switch to convex for higher RTT, as illustrated in Fig. 6(a). For small RTT, the buffers are sufficiently large to maintain peak flow rates, but the limit the amount of data in transit for larger RTT, which is analytically shown to result in a convex profile [15]. Indeed, the small TCP buffers of cluster nodes prior to tuning limited in-transit data, which resulted in the very prominent convex profile in Fig. 6(b).

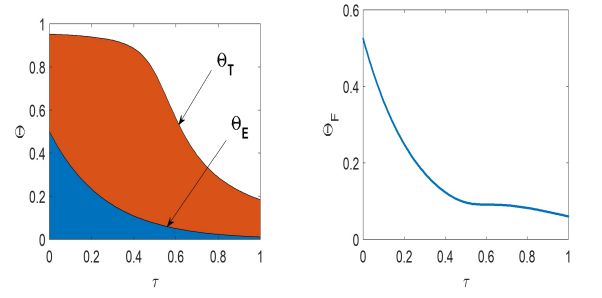
3 THROUGHPUT PROFILES AND COEFFICIENTS

The end-to-end throughput profile $\hat{\Theta}_E(\cdot)$ is composed of the underlying TCP profile $\hat{\Theta}_T(\cdot)$ and the local file throughput at end hosts H_i, $i = 1, 2$, which are quite varied and complex by themselves, as shown in the previous section. For a dedicated connection of capacity L and RTT τ, we have the following boundary conditions:

(i) $\hat{\Theta}_T(\cdot) \leq L$, since TCP throughput is at most and typically lower than the connection capacity L;
(ii) $\hat{\Theta}_E(\cdot) \leq \hat{\Theta}_T(\cdot)$ in most cases, since TCP memory transfers between the hosts are not constrained by the file systems; and



(a) convex-concave coefficient C_{CC} of $\hat{\Theta}_E$ (b) under-utilization coefficient C_U of $\tilde{\Theta}_E$



(c) network $\tilde{\Theta}_T$ and end-to-end $\tilde{\Theta}_E$ (d) file transfer $\tilde{\Theta}_F(\tau)$

Figure 7: Throughput profiles and coefficients

(iii) $\hat{\Theta}_E(\cdot) \leq \min\{\hat{\Theta}_{H_1}, \hat{\Theta}_{H_2}\}$, since the file transfer rates are limited by local file system throughput at both ends of the connection.

In what follows, we first review the key coefficients that have been first proposed in [9] for $\hat{\Theta}_T(\cdot)$ and extend them to end-to-end throughput $\hat{\Theta}_E(\cdot)$. Then, we describe the profile of the file transfer scheme $\hat{\Theta}_F(\cdot)$ and its coefficients.

3.1 Convex-Concave and Utilization-Concavity Coefficients

Let $\hat{\Theta} : [\tau_L, \tau_R] \mapsto [0, L]$ be a generic throughput profile and $\tilde{\Theta} : [0, 1] \mapsto [0, 1]$ be its "normalized" version such that

$$\tilde{\Theta}(\tau) = \frac{1}{L}\hat{\Theta}(\tau)$$

$$\tilde{\Theta}(\tilde{\tau}) = \frac{1}{L}\hat{\Theta}\left(\tau_L + [\tau_R - \tau_L]\,\tilde{\tau}\right),$$

wherein the throughput values are scaled by L, and the operand $\tilde{\tau}$ is translated and scaled from interval $[\tau_L, \tau_R]$ to $[0, 1]$. The *convex-concave coefficient* C_{CC} is the sum of areas above and below the linear interpolation of $\hat{\Theta}(\tau_L)$ and $\hat{\Theta}(\tau_R)$ with positive and negative signs, respectively, as shown in Fig. 7(a). When applied to normalized profile $\tilde{\Theta}$, it has the following simple form [9]

$$C_{CC}\left(\tilde{\Theta}\right) = \left[\int \tilde{\Theta} - \frac{\tilde{\Theta}(\tau_L) + \tilde{\Theta}(\tau_R)}{2}\right] = \int \tilde{\Theta} - \tilde{\Theta}_M,$$

which is the difference between the mean and mid-point $\tilde{\Theta}_M = \frac{\tilde{\Theta}(\tau_L)+\tilde{\Theta}(\tau_R)}{2}$, and takes values in the range $[-\frac{1}{2}, \frac{1}{2}]$. The *under-utilization coefficient* C_U represents the unutilized capacity such that

$$C_U(\tilde{\Theta}) = 1 - \int_0^1 \tilde{\Theta}(\tau)d\tau$$

as shown in Fig. 7(b).

We combine both utilization and convex-concave properties in the *utilization-concavity coefficient* as

$$C_{UC}(\cdot) = \frac{1}{2}\left([1 - C_U(\cdot)] + \left[\frac{1}{2} + C_{CC}(\cdot)\right]\right).$$

In general, it takes values in $[0, 1]$, such that higher values indicate higher utilization and higher concavity level, and will be used in the following sections to compare the performance of different file transfer schemes and methods.

3.2 File Transfer Scheme Profiles

The normalized profile of the file transfer method $\tilde{\Theta}_E(\tau)$ and its underlying TCP profile $\tilde{\Theta}_T(\tau)$ are shown in Fig. 7(c) for a typical case. The gap between them captures the effects of file and storage end-systems, and also the file transfer scheme used to support the transfers between them. Since $\tilde{\Theta}_E(\tau)$ encompasses TCP profile $\tilde{\Theta}_T(\tau)$ and host profiles $\tilde{\Theta}_{H_i}$, $i = 1, 2$, the effectiveness of the file transfer scheme by itself can be assessed by normalizing with respect to the other two profiles as follows. We consider that TCP profile is a non-increasing function of τ, and consider two cases: (a) If file throughput at both end-systems is high enough such that $\tilde{\Theta}_{H_i} \geq \tilde{\Theta}_T(\tau)$, for all $\tau \in [0, 1]$, $i = 1, 2$, we define the throughput profile of the file transfer scheme as $\tilde{\Theta}_F(\tau) = \tilde{\Theta}_E(\tau)/\tilde{\Theta}_T(\tau)$, for $\tau \in [0, 1]$, as illustrated in Fig. 7(d). (b) If the file system at one of sites limits TCP throughput such that $\min\{\tilde{\Theta}_{H_1}, \tilde{\Theta}_{H_2}\} < \tilde{\Theta}_T(0)$, we define

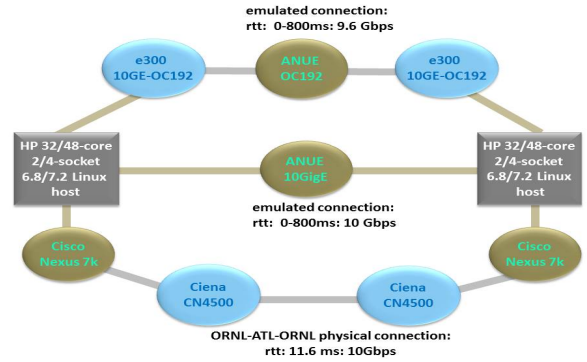
$$\tilde{\Theta}_F(\tau) = \frac{\tilde{\Theta}_E(\tau)}{\tilde{\Theta}_T(\tau)\min\{\tilde{\Theta}_{H_1}, \tilde{\Theta}_{H_2}\}},$$

for $\tau \in [0, 1]$. In both cases, the utilization-concavity coefficient of the file transfer scheme is given by $C_{UC}(\tilde{\Theta}_F)$.

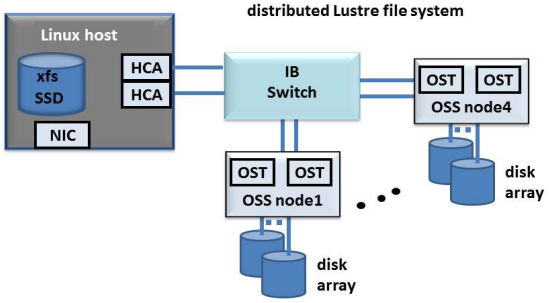
4 NETWORK TESTBED WITH FILE SYSTEMS

Our network testbed consisting of 32-core (`feynman1`, `feynman2`, `tait1` and `tait2`) and 48-core (`bohr05` and `bohr06`) Linux workstations, QDR IB switches, and 10 Gbps hardware connection emulators. For various network connections, hosts with identical configurations are connected in pairs over a back-to-back fiber connection with negligible 0.01 ms RTT and a physical 10GigE connection with 11.6 ms RTT via Cisco and Ciena devices, as shown in Fig. 8(a). We use ANUE devices to emulate 10GigE connections with RTTs $\tau \in \{0.4, 11.8, 22.6, 45.6, 91.6, 183, 366\}$ ms. We chose these RTT values to represent three scenarios of interest: lower values represent cross-country connections, for example, between facilities across the US; 93.6 and 183 ms represent inter-continental connections, for example, between US, Europe, and Asia; and 366 ms represents a connection spanning the globe, which is mainly used as a limiting case.

Memory-to-memory throughput measurements for TCP are collected using *iperf*. Typically, 1-10 parallel streams are used for



(a) Physical and emulated connections between hosts



(b) Lustre and XFS file systems

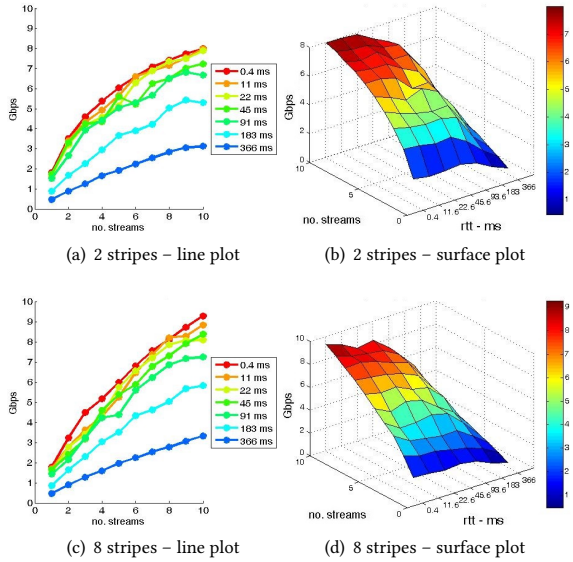
Figure 8: Testbed network connections and file systems

each configuration, and throughput measurements are repeated ten times. TCP buffer sizes are set at largest allowed by the host kernel to avoid TCP-level performance bottlenecks, which for iperf is 2 GB. These settings result in the allocation of 1 GB socket buffer sizes for iperf.

Our testbed consists of a distributed Lustre file system supported by eight OSTs connected over IB QDR switch, as shown in Fig. 8(b). Host systems (`bohrs` and `taits`) are connected to IB switch via HCA and to Ethernet via 10 Gbps Ethernet NICs. In addition, our SSD drives are connected over PCI buses on the hosts `bohr05` and `bohr06`, which mount local XFS file systems. We also consider configurations wherein Lustre is mounted over long-haul connections using LNet routers on `tait1` and `bohr06`, and in this case we utilize IOzone for throughput measurements for both site and remote file systems.

5 FILE TRANSFER MEASUREMENTS

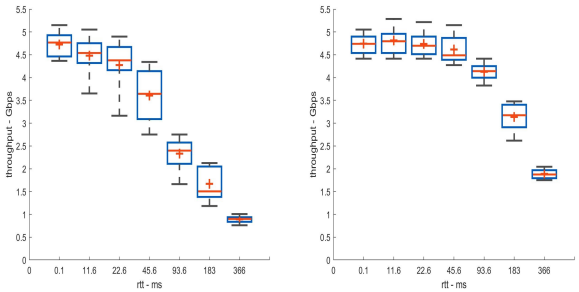
High-performance disk-to-disk transfers between file systems at different sites require the composition of complex file IO and network subsystems, and host orchestration. For example, as mentioned earlier, the Lustre file system employs multiple OSTs to manage collections of disks, multiple OSSs to stripe file contents, and distributed MDSs to provide site-wide file naming and access. However, sustaining high file-transfer rates requires *joint* optimization of subsystem parameters to account for the impedance mismatches



(a) 2 stripes – line plot (b) 2 stripes – surface plot

(c) 8 stripes – line plot (d) 8 stripes – surface plot

Figure 9: Mean direct IO Lustre file write rates



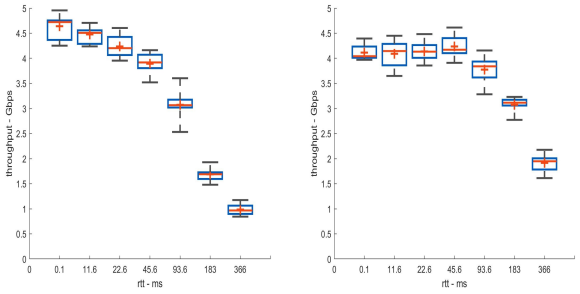
(a) 1 flow (b) 10 flows

Figure 10: GridFTP throughput for Lustre with direct IO

among them [20]. For Lustre file systems, important parameters are the stripe size and number of stripes for the files, and these are typically specified at the creation time; the number of parallel IO threads for read/write operations are specified at the transfer time. To sustain high throughput, IO buffer size and the number of parallel threads are chosen to be sufficiently large, and as we will illustrate, this simple heuristic is not always optimal. For instance, wide-area file transfers over 10 Gbps connections between two Lustre file systems achieve transfer rates of only 1.5 Gbps, when striped across 8 storage servers, accessed with 8 MB buffers, and with 8 IO and TCP threads [16], even though peak network memory-transfer rate and local file throughput are each close to 10 Gbps.

5.1 XDD Measurements

We measured file IO and network throughput and file-transfer rates over Lustre and XFS file systems for a suite of seven emulated connections in the 0-366 ms RTT range (more detailed discussions on these measurements are provided in [16]). We collected two sets of XDD disk-to-disk file transfer measurements, one from XFS to



(a) 1 flow (b) 10 flows

Figure 11: GridFTP throughput for XFS over SSD drives

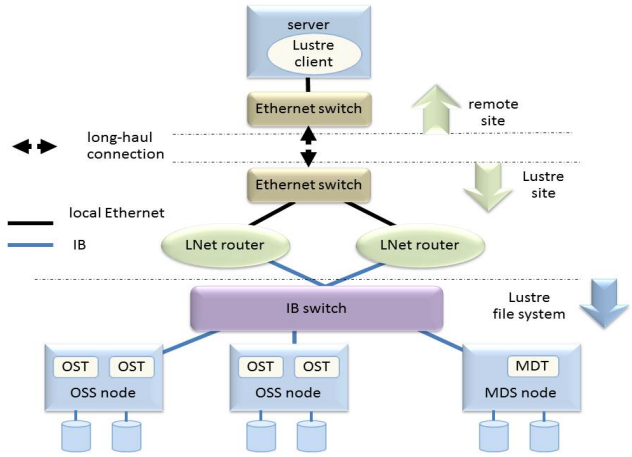
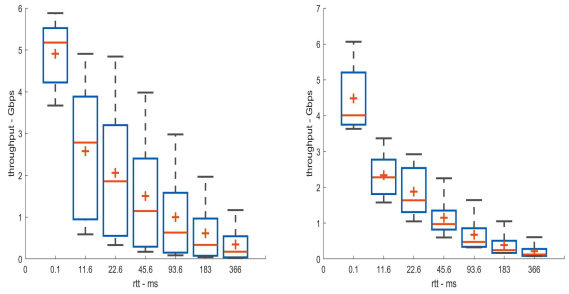


Figure 12: Lustre over long-haul connections using LNet routers between local IB and Ethernet

XFS and one from Lustre to Lustre. We considered both buffered IO (the Linux default) and direct IO options for Lustre. In the latter, XDD avoids the local copies of files on hosts by directly reading and writing into its buffers, which significantly improves the transfer rates. The mean file write throughput plots, using direct IO Lustre and either 2 or 8 stripes, are shown in Fig. 9. Results based on these and other measurements are summarized in [16]: (a) strategies of large buffers and higher parallelism do not always translate into higher transfer rates; (b) direct IO methods that avoid file buffers at the hosts provide higher wide-area transfer rates, and (c) significant statistical variations in measurements, due to complex interactions of non-linear TCP dynamics with parallel file IO streams, necessitate repeated measurements to ensure confidence in inferences based on them.

5.2 GridFTP Measurements

We collected measurements using GridFTP under the same configurations used for XDD measurements. Transfers of 10 GB files are carried out between the bohrs, where the buffer sizes are again set to be the largest allowable value and each experiment is repeated 10 times. Figs. 10 and 11 show the boxplots that illustrate GridFTP throughput performance over eight stripe direct Lustre and XFS



(a) 48-core stand-alone hosts (b) 32-core cluster nodes

Figure 13: Throughput profiles of wide-area Lustre

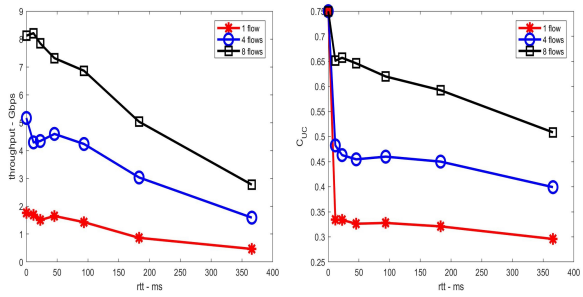
file systems respectively, with 1 and 10 flows. From these plots, one can easily observe that the overall throughput profiles of GridFTP are somewhat lower compared to XDD (e.g., with a peak throughput below 5.5 Gbps); in addition, using more flows does not lead to significant increases in throughput across all RTTs, although throughputs appear more sustained with increasing RTTs under higher flow counts, resulting in somewhat more concave profile compared to the obviously convex profile under the single-flow configuration.

5.3 Lustre with LNet Routers

We utilize Lustre Ethernet clients on remote servers to mount file system over wide-area networks as shown in Fig. 12. At the server site, Lustre IB clients on host systems are used to connect OSS over site IB network. Some of these hosts are also connected to Ethernet Local-Area Network (LAN), which in turn is connected to WAN. These hosts are configured as LNet routers that route between IB and Ethernet. The remote hosts are connected to these hosts over the wide-area Ethernet connections. The Lustre file system is mounted at remote hosts over Ethernet such that it is functionally similar to a local Lustre system. While this configuration provides complete Lustre functionality at remote sites, file operations are performed over data paths composed of Ethernet wide-area connections and site IB connections. The data transport over wide-area connections is controlled by TCP and that over site connections is controlled by IB protocol. Consequently, the file read and write performance depends on TCP and IB parameters as well as Lustre parameters, such as LNet buffers and peer credits. Indeed these parameters must be jointly "tuned" to avoid performance bottlenecks, as will be described subsequently.

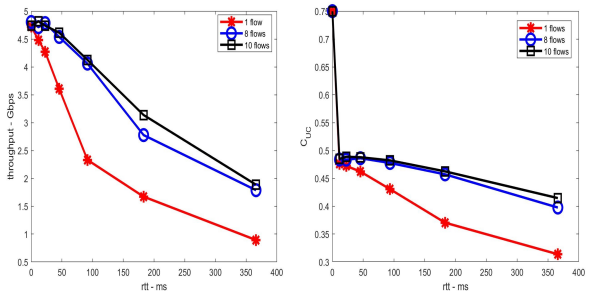
Throughput measurements using IOzone writes are shown in Fig. 13 for 10GigE emulated connections, where each measurement is repeated 10 times at each RTT value. These measurements establish that peak Lustre throughput of 1 Gbps can indeed be achieved over connections with 366 ms RTT as shown in Fig. 13(a).

The Lustre throughput profiles are overall lower than the corresponding iperf profiles, which indicates that Lustre parameters are dominant in determining the throughput values as well as the shape of throughput profiles. The difference between the throughput of two types of hosts is within 10%, indicating that it is primarily determined by Lustre parameters. More importantly, it is striking



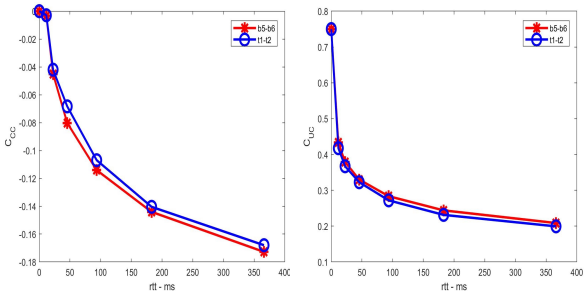
(a) throughput (b) C_{UC}

Figure 14: XDD transfer analytics: direct IO Lustre, 8 stripes



(a) throughput (b) C_{UC}

Figure 15: GridFTP transfer analytics: direct IO Lustre, 8 stripes



(a) C_{CC} (b) C_{UC}

Figure 16: LNet router analytics: Lustre with 8 stripes

that these profiles are convex, which is indicative of IO limits due to Lustre file system. Two classes of Lustre LNet parameters are critical to throughput performance: (i) LNet buffers, which limit the amount of data that can be read or written by Lustre clients, and (ii) peer credits for IB and Ethernet connections, which limit the amount of in-transit data between Lustre clients and servers. We increased buffer sizes from default 65 KB to 2 GB, which only resulted in a small improvement in throughput, namely within 10%. Thus, these buffer sizes are not the main contributors to the convex throughput profile. Our conjecture is that these buffers are not getting filled during the transfers due to current peer credits limit of 256. These credits must be increased separately for IB and Ethernet, which in turn requires multiple configuration changes to Lustre installation, namely, clients, LNet routers and servers. This task and testing of resultant performance will be carried out as a part of future work.

6 FILE TRANSFER ANALYTICS

In this section, we first analyze the end-to-end file transfer method throughput $\hat{\Theta}_E$ using the utilization and concavity coefficients described in Section 3. Then, we focus on the effect of the file transfer scheme throughput $\hat{\Theta}_F$, namely, GridFTP, XDD and LNet, by normalizing $\hat{\Theta}_E$ with respect to the corresponding TCP iperf measurements that constitute $\hat{\Theta}_T$. The normalized versions enable us to objectively compare both the utilization and concavity of various profiles, namely, network, end-to-end, and file transfer scheme throughputs, using C_{UC}: its values closer to 1.0 represent high utilization and concave profile, those closer to 0 indicate lower utilization and convexity, and 0.5 represents a class of linear profiles, of which strictly linear profile is a special case.

6.1 End-to-End Transfer Method Analytics: $\hat{\Theta}_E$

Fig. 14 shows the throughput of XDD direct IO file transfers and their corresponding C_{UC} curves under various RTTs and flow counts. The utilization-concavity coefficient C_{UC} provides a stabilized account of the proportion of utilized link capacity, and it largely follows the trend of the throughput profile curve. For example, with 8 flows, the coefficient stays above 0.5 for all RTTs, demonstrating a much improved performance over the single-flow case, where the coefficient are only above 0.3 even for lower RTTs. Meanwhile, Fig. 15 shows the GridFTP file transfer performance analytics with 8 stripe direct IO using 1, 8, and 10 flows. Compared to the previous case, here the peak throughput is below 5 Gbps and decreases with RTT, which is reflected by utilization coefficient being below 0.5.

For LNet Lustre transfers analytics plots shown in Fig. 16, using either of (`bohr` or `taits`) pairs yields similar all-negative C_{CC} curves, reflecting the convex profiles. This convexity is primarily a result of IO and Ethernet credits of LNet that only partially filled the buffers, and indeed increasing LNet buffers did not improve the performance [14]. This low utilization and convexity of LNet router transfers are indicated by C_{UC} values below 0.5, reflecting the highly convex profile as seen previously in Fig. 13.

At the highest RTT, C_{UC} values for XDD XFS transfers are highest among these cases, and those of LNet router are the lowest; in between these two are the XDD with direct IO Luster, GridFTP using direct IO Lustre, followed by GridFTP for XFS, as shown in Fig. 2. Thus, the utilization-concavity coefficient C_{UC} is an objective indicator of the performance of file transfer methods, which takes into account the peak and concavity of throughput profiles.

6.2 GridFTP, XDD and LNet Analytics: $\hat{\Theta}_F$

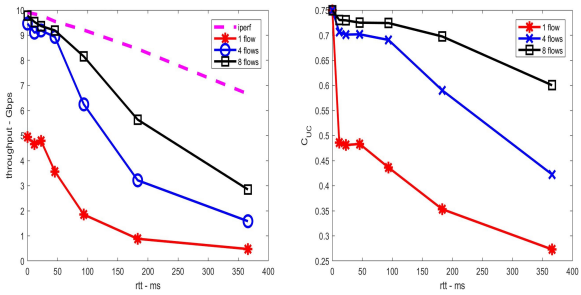
The profile $\tilde{\Theta}_F(\tau) = \tilde{\Theta}_E(\tau)/\tilde{\Theta}_T(\tau)$ represents the effectiveness of file transfer scheme relative to TCP memory transfers, when host throughput is not a limitation. Table 1 presents the concavity coefficient $C_{CC}(\tilde{\Theta}_F)$ and utilization coefficient $C_{UC}(\tilde{\Theta}_F)$ for XFS files

Table 1: $C_{CC}(\tilde{\Theta}_F)$ and $C_{UC}(\tilde{\Theta}_F)$ for XDD file transfer performance with XFS and 8 parallel flows

RTT (ms)	0.4	11.8	22.6	45.6	91.6	183	366
point mean file transfer throughput $\hat{\Theta}_E$ (Gbps)	9.784	9.532	9.368	9.191	8.152	5.636	2.849
point mean iperf throughput $\hat{\Theta}_T$	9.905	9.850	9.793	9.553	9.208	8.454	6.660
point normalized throughput $\tilde{\Theta}_F$	0.988	0.968	0.957	0.962	0.885	0.667	0.428
interval mean of normalized throughput $\bar{\tilde{\Theta}}_F$	1.000	0.969	0.966	0.963	0.943	0.861	0.704
interval midpoint of normalized throughput $\tilde{\Theta}_{M,F}$	1.000	0.978	0.972	0.975	0.937	0.827	0.708
$C_{CC}(\tilde{\Theta}_F)$	0	−0.0084	−0.0064	−0.0124	0.0061	0.0340	−0.0036
$C_{UC}(\tilde{\Theta}_F)$	0.7500	0.7304	0.7297	0.7251	0.7244	0.6976	0.6003

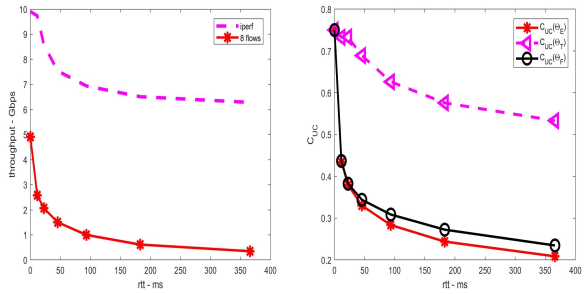
Table 2: $C_{UC}(\tilde{\Theta}_F)$ for IOzone file transfer performance with LNet Lustre and 8 stripes

RTT (ms)	0.1	11.8	22.6	45.6	91.6	183	366
mean file transfer throughput $\hat{\Theta}_E$ (Gbps)	4.908	2.578	2.063	1.502	0.998	0.614	0.349
normalized end-to-end profile $\tilde{\Theta}_E$	0.491	0.258	0.206	0.150	0.100	0.061	0.035
$C_{UC}(\tilde{\Theta}_E)$	0.7500	0.4339	0.3792	0.3300	0.2835	0.2440	0.2087
point mean iperf throughput $\hat{\Theta}_T$	9.908	9.735	8.581	7.535	6.937	6.510	6.283
normalized TCP profile $\tilde{\Theta}_T$	0.991	0.974	0.858	0.754	0.694	0.651	0.628
$C_{UC}(\tilde{\Theta}_T)$	0.7500	0.7326	0.7333	0.6890	0.6263	0.5758	0.5332
normalized file system profile $\tilde{\Theta}_F$	0.495	0.265	0.240	0.199	0.144	0.094	0.056
$C_{UC}(\tilde{\Theta}_F)$	0.7500	0.4368	0.3825	0.3440	0.3087	0.2725	0.2347



(a) throughput profiles (b) $C_{UC}(\tilde{\Theta}_F)$

Figure 17: XDD transfer with XFS over SSD



(a) throughput profiles (b) C_{UC}

Figure 18: IOzone measurements for LNet extended Lustre

transfers using XDD scheme; these are slightly higher compared to their end-to-end counterparts, especially at higher RTTs due to the decreasing $\tilde{\Theta}_T$ as shown in Fig. 17(b). The normalized throughput and C_{UC} values for LNet scheme for Lustre file transfer measurements between bohrs shown in Table 2 are limited by LNet credits [14], and hence are lower than those in the above XDD cases. Compared to the nearly linear drop in throughput with increasing RTT in Fig. 17(a), in Fig. 18(a), the iperf throughput decreases more rapidly starting around intermediate RTTs, resulting in much lower $\tilde{\Theta}_T$ and $C_{UC}(\tilde{\Theta}_T)$ values; however, since the end-to-end transfer throughput values are also much lower to begin with, the resulting $\tilde{\Theta}_F$ is still highly convex and the improvement of $C_{UC}(\tilde{\Theta}_F)$ over $C_{UC}(\tilde{\Theta}_E)$ is far from significant as shown in Fig. 18(b).

7 CONFIDENCE ESTIMATES

The results in previous sections depend critically on estimates of $C_{UC}(\hat{\Theta})$ in making performance inferences. These estimates are subject to randomness inherent in measurements as a result of the non-linearity of transport dynamics interacting with complex host, file, and I/O systems. We now derive confidence bounds for estimates of $C_{UC}(\hat{\Theta})$, which show their statistical soundness. Let Θ_τ be a random variable representing the throughput of a connection in $\mathcal{S}$ with RTT $\tau \in \mathcal{S}_\tau$ such that $\bar{\Theta}(\tau) = \int \Theta_\tau dP_{\Theta_\tau}$ is the expected throughput. The *regression performance profile* $\bar{\Theta}$ is approximated by $\hat{\Theta}$ using available measurements. However, $\bar{\Theta}$ and its coefficients depend on the joint distribution P_{Θ_τ}, which is complex as it depends on TCP dynamics over the connection, file I/O at the host, and the interactions between the two. We now show that C_{UC} of $\hat{\Theta}$ is close to that of the ideal $\bar{\Theta}$ with a probability that improves with the number of measurements independent of the complex underlying distributions. The expected error in an estimate f of the regression function is defined as

$$I(f) = \int [f(\tau) - \Theta_\tau]^2 \, dP_{\Theta_\tau, \tau}.$$

We are given l independently and identically distributed throughput measurements $\Theta_{\tau_1}, \Theta_{\tau_2}, \ldots, \Theta_{\tau_l}$. The estimator $\hat{\Theta}$ minimizes

the empirical error, defined for an estimate f as

$$\hat{I}(f) = \frac{1}{l}\sum_{i=1}^{l}\left[f(\tau_i) - \Theta_{\tau_i}\right]^2.$$

Then there exists a confidence function $\delta(\cdot)$ such that

$$P\left\{I(\hat{\Theta}) - I(\bar{\Theta}) > \epsilon\right\} \leq \delta(\epsilon, l),$$

which shows that $\hat{\Theta}$ achieves optimal error within ϵ and with confidence δ, which improves with the number of measurements [15]. First, we have

$$|C_{U,C}(\hat{\Theta}) - C_{U,C}(\bar{\Theta})| \leq |I(\hat{\Theta}) - I(\bar{\Theta})| + |C_{C,C}(\hat{\Theta}) - C_{C,C}(\bar{\Theta})|.$$

Then, we consider the following bound

$$P\left\{\max_{f\in\mathcal{F}}|I(f) - \hat{I}(f)| > \epsilon/2\right\} \leq \delta(\epsilon, l),$$

where $\mathcal{F}$ is a class of functions of bounded total variation [2], and an explicit formula for $\delta(\epsilon, l)$ can be found in [13]. By combining these two results, we have

$$P\left\{|C_{U,C}(\hat{\Theta}) - C_{U,C}(\bar{\Theta})| > \epsilon\right\} \leq 2\delta(\epsilon/2, l).$$

which shows that the estimate $C_{U,C}(\hat{\Theta})$ of previous sections is a sound estimate of true $C_{U,C}(\bar{\Theta})$. This confidence bound $2\delta(\epsilon/2, l)$ is distribution-free in that it is valid under all underlying distributions P_{Θ_τ}, and improves monotonically with the number of measurements l.

8 CONCLUSIONS

We presented a study of throughput measurements and analytics of wide-area file transfers needed by distributed science and big data computations. Extensive file throughput measurements are collected over dedicated 10 Gbps connections using GridFTP and XDD file transfer tools, and Lustre file system extended using LNet routers. The file throughput measurements were quite varied due to the complexities of host, file, IO, and disk systems, and their interactions, which make their comparison and performance optimization very challenging. We presented unifying analytics based on the convexity-concavity geometry of throughput profiles, and proposed using the utilization-concavity coefficient to characterize the overall file transfer performance. Our results provided guidelines for performance optimizations by highlighting the significant roles of buffer sizes and utilization, and parallelism implemented by file transfer methods.

Further investigations, including additional test configuration and examination of additional configurations, are needed to further improve throughput performance over shared connections. It would be of future interest to extend the calculus of throughput profiles described here in two directions, namely, deeper to focus on subsystems and broader to encompass data transfer infrastructures.

Acknowledgments

This work is funded by RAMSES project and the Applied Mathematics Program, Office of Advanced Computing Research, U.S. Department of Energy (DOE), and by Extreme Scale Systems Center, sponsored by U. S. Department of Defense, and performed at Oak Ridge National Laboratory managed by UT-Battelle, LLC for DOE under Contract No. DE-AC05-00OR22725.

REFERENCES

[1] W. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu, and I. Foster. The Globus striped GridFTP framework and server. In *ACM/IEEE Conference on Supercomputing*, pages 54–64, Washington, DC, 2005. IEEE Computer Society.

[2] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

[3] Aspera transfer service, accessed March 28, 2018. http://asperasoft.com.

[4] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *Intl J. Supercomputer Applications*, 11(2):115–128, 1997.

[5] Y. Gu and R. L. Grossman. UDT: UDP-based data transfer for high-speed wide area networks. *Computer Networks*, 51(7), 2007.

[6] R. Henschel, S. Simms, D. Hancock, S. Michael, T. Johnson, N. Heald, T. William, et al. Demonstrating Lustre over a 100Gbps wide area network of 3,500km. In *Internationl Conference on High Performance Computing, Networking, Storage and Analysis*, pages 1–8, Nov. 2012.

[7] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat. B4: Experience with a globally-deployed software defined WAN. *SIGCOMM Comput. Commun. Rev.*, 43(4):3–14, Oct. 2013.

[8] T. Kelly. Scalable TCP: Improving performance in high speed wide area networks. *Computer Communication Review*, 33(2):83–91, 2003.

[9] Q. Liu and N. S. V. Rao. On concavity and utilization analytics of wide-area network transport protocols. In *Proc. 20th IEEE Conference on High Performance Computing and Communications (HPCC)*, Exeter, U.K., Jun. 2018.

[10] Multi-core aware data transfer middleware, accessed March 28, 2018. mdtm.fnal.gov.

[11] S. Michael, L. Zhen, R. Henschel, S. Simms, E. Barton, and M. Link. A study of Lustre networking over a 100 gigabit wide area network with 50 milliseconds of latency. In *5th International Workshop on Data-Intensive Distributed Computing*, page 43âĂŞ52, 2012.

[12] On-demand Secure Circuits and Advance Reservation System. http://www.es.net/oscars.

[13] N. S. V. Rao. SDN solutions for switching dedicated long-haul connections: Measurements and comparative analysis. *International Journal on Advances in Networks and Services*, 9(3-4), 2016.

[14] N. S. V. Rao, N. Imam, J. Hanley, and O. Sarp. Wide-area Lustre file system using LNet routers. In *12th Annual IEEE International Systems Conference*, 2018.

[15] N. S. V. Rao, Q. Liu, S. Sen, J. Henley, I. T. Foster, R. Kettimuthu, D. Towsley, and G. Vardoyan. TCP throughput profiles using measurements over dedicated connections. In *ACM Symposium on High-Performance Parallel and Distributed Computing*, Washington, DC, July-August. 2017.

[16] N. S. V. Rao, Q. Liu, S. Sen, G. Hinkel, N. Imam, B. W. Settlemyer, I. T. Foster, et al. Experimental analysis of file transfer rates over wide-area dedicated connections. In *18th IEEE International Conference on High Performance Computing and Communications (HPCC)*, pages 198–205, Sydney, Australia, Dec. 2016.

[17] N. S. V. Rao, Q. Liu, S. Sen, D. Towsley, G. Vardoyan, I. T. Foster, and R. Kettimuthu. Experiments and analyses of data transfers over wide-area dedicated connections. In *26th International Conference on Computer Communications and Network*, 2017.

[18] N. S. V. Rao, D. Towsley, G. Vardoyan, B. W. Settlemyer, I. T. Foster, and R. Kettimuthu. Sustained wide-area TCP memory transfers over dedicated connections. In *IEEE International Conference on High Performance and Smart Computing*, New York, NY, Aug. 2015.

[19] S. Sen, N. S. V. Rao, Q. Liu, N. Imam, I. T. Foster, and R. Kettimuthu. On analytics of file transfer rates over dedicated wide-area connections. In *First International Workshop on Workflow Science (WOWS)*, Auckland, New Zealand, October 2017. in conjunction with 13th IEEE International Conference on e-Science.

[20] B. W. Settlemyer, J. D. Dobson, S. W. Hodson, J. A. Kuehn, S. W. Poole, and T. M. Ruwart. A technique for moving large data sets over high-performance long distance networks. In *IEEE 27th Symposium on Mass Storage Systems and Technologies*, pages 1–6, May 2011.

[21] R. N. Shorten and D. J. Leith. H-TCP: TCP for high-speed and long-distance networks. In *3rd International Workshop on Protocols for Fast Long-Distance Networks*, 2004.

[22] GT 4.0 GridFTP. http://www.globus.org.

[23] XDD – The eXtreme dd toolset, accessed March 28, 2018. https://github.com/bws/xdd.