Learning Concave-Convex Profiles of Data Transport Over Dedicated Connections *

Nageswara S. V. Rao[1], Saytabrata Sen[1], Zhengchun Liu[2], Rajkumar Kettimuthu[2], and Ian Foster[2]

[1] Oak Ridge National Laboratory, Oak Ridge, TN USA
{raons,sens}@ornl.gov
[2] Argonne National Laboratory, Argonne, IL USA
{zhengchun.liu,kettimut,foster}@anl.gov

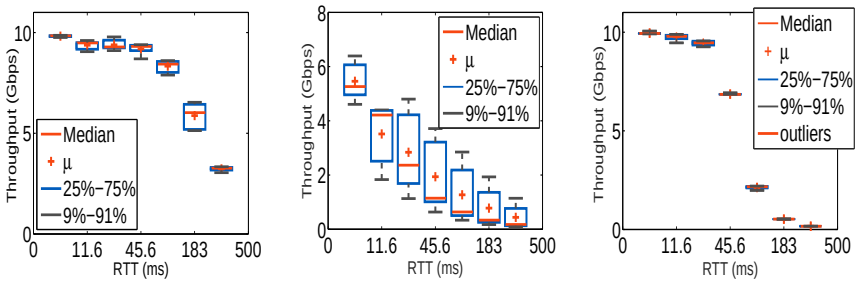
Abstract. Dedicated data transport infrastructures are increasingly being deployed to support distributed big-data and high-performance computing scenarios. These infrastructures employ data transfer nodes that use sophisticated software stacks to support network transport among sites that often house distributed file and storage systems. Throughput measurements collected over such infrastructures for a range of round trip times (RTTs) reflect the underlying complex end-to-end connections, and have revealed dichotomous throughput profiles as functions of RTT. In particular, concave regions at lower RTTs indicate near-optimal performance, and convex regions at higher RTTs indicate bottlenecks due to factors such as buffer or credit limits. We present a machine learning method that explicitly infers these concave and convex regions and transitions between them using sigmoid functions. We also provide distribution-free confidence estimates for the generalization error of these concave-convex profile estimates. Throughput profiles for data transfers over 10 Gbps connections with 0–366 ms RTT provide important performance insights, including the near optimality of transfers performed with the XDD tool between XFS filesystems, and the performance limits of wide-area Lustre extensions using LNet routers. A direct application of generic machine learning packages does not adequately highlight these critical performance regions or provide as precise confidence estimates.

Keywords: Data Transport · Throughput Profile · Concavity-Convexity · Generalization Bounds

1 Introduction

There have been unprecedented increases in the volume and types of data transfers over long-distance network connections in a number of scenarios, such as transfers of partial computations over geographically dispersed cloud-server sites for big data

*This work is funded by RAMSES project and Applied Mathematics program, Office of Advanced Computing Research, U.S. Department of Energy, 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 U.S. Department of Energy under Contract No. DE-AC05-00OR22725.



(a) XDD: XFS on SSD (b) LNet-routed Lustre (c) memory: Scalable TCP

Fig. 1: Throughput measurements over dedicated 10GigE connections for RTT $\in [0, 366]$ ms: (a) concave profile of near-optimal XDD transfers, (b) convex profile due to LNet router limits, and (c) intermediate concave-convex profile of memory transfer.

computations. Extensive throughput measurements collected over a range of testbed and production infrastructures (for example, [9, 19]) show that performance can vary greatly as a result of both transfer characteristics and choices made along four dimensions:

(i) Data Transfer Node (DTN) host systems, which can vary significantly in terms of the number of cores, Network Interface Card (NIC) capability, and connectivity;
(ii) file and disk systems, such as Lustre [10], GPFS [5], and XFS [26], installed on Solid State Disk (SSD) or hard disk arrays;
(iii) network protocols, for example, CUBIC [20], H-TCP [23], and BBR [4] versions of Transmission Control Protocol (TCP); and
(iv) file transfer software, such as Globus [2], GridFTP [1], XDD [25, 22], UDT [6], MDTM [11], Aspera [3], and LNet extensions of Lustre [16].

Thus throughput measurements need to be analyzed systematically to reveal performance trends of the components and infrastructures.

We denote the throughput at time t over a connection of RTT τ as $\theta(\tau, t)$, for a given configuration of end-to-end connection. We call its expectation over an observation period T_O the *throughput profile* (a function of τ), given by

$$\Theta_A(\tau) = \frac{1}{T_O} \int_0^{T_O} \theta(\tau, t)\, dt$$

where modality $A = T$ and $A = E$ correspond to memory transfers using TCP and end-to-end disk file transfers, respectively. Each modality embodies the combined effects of corresponding components and their configurations; for file transfers, it reflects the composition of filesystems, network connections, and their couplings through host buffers. In general, $\Theta_A(\tau)$ is a random quantity and its distribution $\mathbf{P}_{\Theta_A(\tau)}$ is complex, since it depends on the properties and instantaneous states of network connections, filesystems, and transfer hosts. Throughput measurements are used to estimate its approximation $\hat{\Theta}_A(\tau)$, in order to characterize transport performance as RTT is varied.

To illustrate the importance of throughput profile properties, we show in Figs. 1(a)-(c) "measured" throughput profiles for three scenarios: (a) XDD transfers between sites

with XFS filesystems mounted on SSD storage, (b) file copy operations between Lustre filesystems extended over wide-area connections using LNet routers, and (c) memory transfers between transfer hosts with identical configurations. Data transfers in these scenarios are over dedicated 10GigE connections with RTT $\tau \in [0, 366]$ ms. The profiles in Figs. 1(a) and 1(b) are quite different not only in their peak throughput values (10 and 4.5 Gbps respectively) but also in their *concave* and *convex* shapes, respectively. The first represents a near-optimal performance achieved by balancing and tuning XFS and TCP parameters in XDD [18], whereas the second represents a performance bottleneck due to LNet credit limits [16]. On the other hand, the third profile, in Fig. 1(c), shows a combination of both concave and convex regions [17], wherein TCP buffers limit throughput values beyond a certain RTT at which the profile switches from concave to convex in shape.

From the perspective of network performance, a concave region is highly desirable since intermediate-RTT throughput values are, by definition, higher than the linear interpolation; this is often an indicator of near-optimal throughput performance. Now, obtaining a concave throughout profile requires (i) selection of TCP version and transport method parameters, (ii) selection of file, I/O, and storage system parameters, and (iii) joint optimization of these parameters as well as the interactions between the complex subsystems. On the other hand, in a convex region, intermediate-RTT throughput values can only be guaranteed to be higher than the minimum observed; this is often an indicator of system/component limits, for example, the LNet credits and TCP buffer sizes in Figs. 1(b) and 1(c), respectively. Furthermore, the shape of $\hat{\Theta}_A(\tau)$ has a deeper connection to the time dynamics of data transfers [17, 8]. For example, a concave profile requires a fast ramp-up followed by stable throughput rates, which in turn requires that file I/O and network connections be matched. Thus, profile estimates that accurately reflect concave and convex regions are highly insightful for performance prediction and diagnosis.

To estimate the concave-convex profile regions, we present a machine learning method based on the flipped sigmoid function $g_{a_1,\tau_1}(\tau) = 1 - \frac{1}{1+e^{-a_1(\tau-\tau_1)}}$. We use this method to estimate, given a set of performance measurements, a throughput profile using the function $f_{\Theta_A}(\tau)$, given by

$$f_{\Theta_A}(\tau) = b\,[g_{a_1,\tau_1}(\tau) I(\tau \leq \tau_T) + g_{a_2,\tau_2}(\tau) I(\tau \geq \tau_T)]$$

where $I(\cdot)$ is the indicator function, $g_{a_1,\tau_1}(\tau)$ is the concave part, and $g_{a_2,\tau_2}(\tau)$ is the convex part. We apply this method to extensive throughput measurements over 10 Gbps connections with 0-366 ms RTT for:

(i) memory transfers using different TCP configurations,
(ii) file transfers between Lustre and XFS filesystems using XDD and GridFTP, and
(iii) file copy operations using Lustre filesystems mounted over wide-area connections using LNet routers.

In addition, we show analytically that the estimation of this concave-convex profile function is statistically sound [24] in that its expected error is close to optimal, with a probability that improves with the number of measurements. This guarantee is independent of the composite distribution $\mathbf{P}_{\Theta_A(\tau)}$ that depends on various hardware and software components and complex interactions between them.

To further evaluate our machine learning method, we also apply some generic machine learning modules from Python and MATLAB libraries to estimate the throughput profiles. We find that these methods do not, in most cases, adequately capture the critical concave-convex regions of the throughput profiles. The confidence probability bounds for their generalization errors are not readily available, but can be derived by using properties such as bounded variance or smoothness [15, 21]. The generalization bounds for our sigmoid-based method use the Lipschitz property and are sharper, as they require only a two-dimensional metric cover, rather than the higher-dimensional covers needed in generic learning techniques.

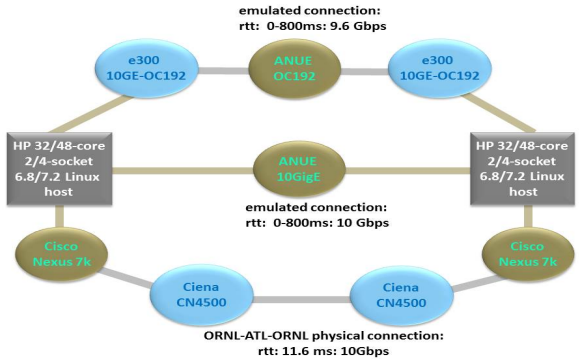
The rest of this paper is as follows. A brief description of our testbed used for measurements is provided in Section 2. The concave and convex regions of memory and file transfer throughput profiles are discussed in Section 3, along with the proposed profile-estimate $f_{\Theta_A}(\tau)$ computed based on measurements. A confidence probability estimate for generalization error is derived in Section 4. Application of some generic machine learning modules is described in Section 5. Conclusions and open areas are briefly described in Section 6.

2 Testbed and Measurements

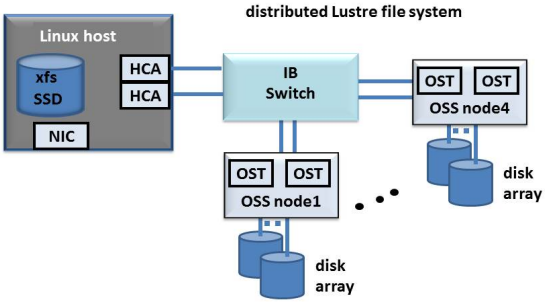
We collected the measurements used in this paper on a testbed consisting of multiple data transfer servers, 10 Gbps wide-area hardware connection emulators, and a distributed Lustre filesystem with LNet routers. The testbed consists of 32-core (`feynman1`, `feynman2`, `tait1`, and `tait2`) and 48-core (`bohr05` and `bohr06`) Linux workstations, QDR Infiniband (IB) switches, and 10 Gbps Ethernet switches. 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. 2(a). We use ANUE devices to emulate (in hardware) 10GigE connections with RTTs $\tau \in [0, 366]$ ms. These RTT values are strategically chosen to represent a global infrastructure with three scenarios of interest: (a) smaller values represent cross-country connections, for example, facilities distributed across the US; (b) 93.6 and 183 ms represent inter-continental connections, for example, among US, Europe, and Asia; and (c) 366 ms represents a connection spanning the globe, which is mainly used as a limiting case.

The Lustre filesystem is supported by eight OSTs connected over IB QDR switch, as shown in Fig. 2(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 filesystems. We also consider configurations wherein Lustre is mounted over long-haul connections using LNet routers on `tait1` and `bohr06`.

Memory-to-memory throughput measurements for TCP are collected using *iperf*. Typically, 1-10 parallel streams are used for each configuration, and throughput measurements are repeated 10 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. File transfers between the sites and different filesystems are carried out using XDD and GridFTP which pro-



(a) physical and emulated connections between hosts



(b) Lustre and XFS filesystems

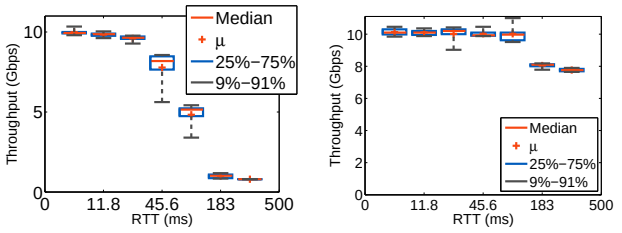
Fig. 2: Testbed network connections and filesystems.

vide the throughput measurements. For Lustre over wide-area connections, throughput measurements are made using IOzone executed on a host with Lustre Ethernet clients that access remote IB Lustre system via LNet routers.

3 Throughput Profiles: Convexity-Concavity

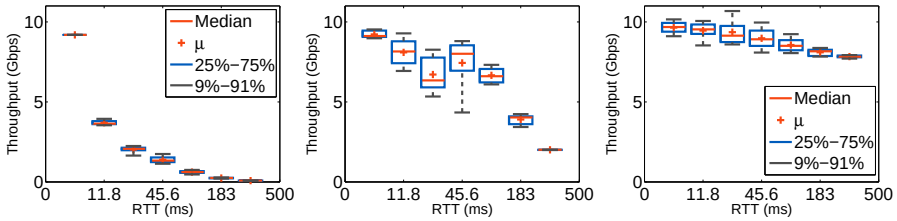
Using memory and file transfer measurements collected over connections with RTT τ_k, $k = 1,2,\ldots,n$, we derive the estimate $\hat{f}_{\Theta_A}(\tau)$ of $\Theta_A(\tau)$. The flipped sigmoid function g_{a,τ_a} is concave for $\tau \leq \tau_a$ and switches to convex for $\tau \geq \tau_a$. The condition $\tau_{a_2} \leq \tau_T \leq \tau_{a_1}$ ensures the concave and convex regions to the left and right of the *transition-RTT* respectively, as illustrated in Fig. 5. Let $\theta(\tau_k, t_i^k)$ denote ith throughput measurement collected at RTT τ_k, $k = 1,2,\ldots,n$, and at time t_i^k, $i = 1,2,\ldots,n_k$. We scale all measurements by the connection capacity such that $b = 1$. We estimate the parameters a_1, τ_{a_1}, a_2, τ_{a_2}, and the transition-RTT τ_T by minimizing the *sum-squared error* of the fit $f_{\Theta_A}(\tau)$ based on measurements, which is defined as

$$\hat{S}(f_{\Theta_A}) = \sum_{\tau_k \leq \tau_T} \sum_{i=1}^{n_k} \left(\theta(\tau_k, t_i^k) - g_{a_1,\tau_{a_1}}(\tau_k)\right)^2 + \sum_{\tau_k \geq \tau_T} \sum_{i=1}^{n_k} \left(\theta(\tau_k, t_i^k) - g_{a_2,\tau_{a_2}}(\tau_k)\right)^2, \qquad (3.1)$$



(a) $f1\_10gige\_f2$, 1 stream (b) $f1\_10gige\_f2$, 10 streams

Fig. 3: Throughput profiles improve with number of parallel streams.



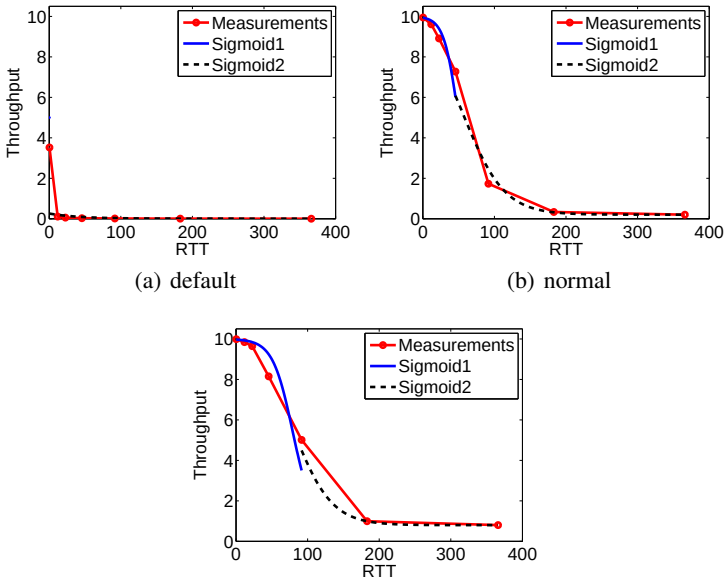
(a) default (b) normal (c) large

Fig. 4: Throughput profiles improve with larger buffer sizes.

where the parameters are bounded such that $a_1, a_2 \in [-A, A]$ and $\tau_{a_1}, \tau_{a_2} \in [-T, T]$, and $\tau_T \in \{\tau_k : k = 1, 2, \ldots, n\}$.

3.1 Various Throughput Profiles

The three different throughput profile shapes illustrated in Fig. 1 correspond to scenarios with disparate end-subsystems and configurations. However, profiles may vary significantly even when end-subsystems are identical, as a result of different transport parameters. For example, Fig. 3, which presents throughput measurements for memory transfers using CUBIC with large buffers, shows that more streams not only increase the aggregate throughput, but also extend the concave region; here, the convex region with a single stream mostly disappears with 10 streams. TCP/IP buffer sizes have a similar effect, with larger buffers both increasing mean throughput and extending the concave region. As seen from Fig. 4, for CUBIC with 10 streams, the default buffer size results in an entirely convex profile; with the normal buffer size recommended for 100 ms RTT, a concave region (leading up to 91.6 ms) is followed by a convex region; finally, a large buffer extends the concave region beyond 183 ms. We will show next that $\hat{f}_{\Theta_A}(\tau)$ estimated using measurements provides us a systematic way to identify the concave and convex regions, and in particular, all convex regions that indicate performance bottlenecks and potential improvements.



(a) default (b) normal

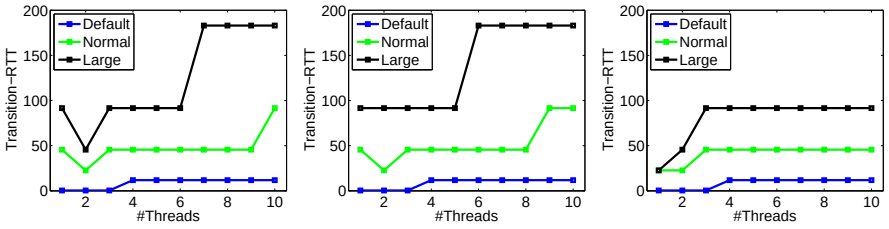
(c) large

Fig. 5: Sigmoid-regression fits of throughput profiles with various buffer sizes for single CUBIC stream.

3.2 Estimates for Memory Transfers

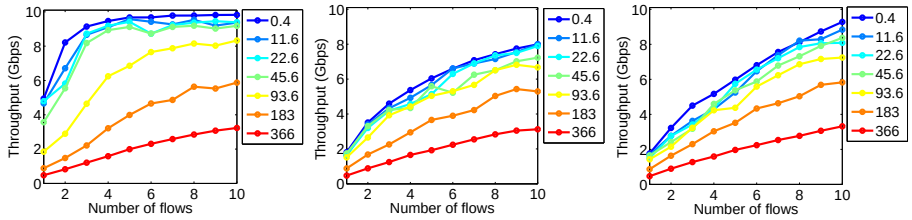
The concave-convex fit $\hat{f}_{\Theta_T}(\tau)$ for single stream CUBIC measurements over 10GigE connections for three different buffer sizes are shown in Fig. 5 along with the measurements. As in previous section with 10 streams, the profile is entirely convex at the default buffer size, and consequently there is only a convex portion to the sigmoid fit. For normal and large buffer sizes, both the concave and convex sigmoid fits are present, as shown with solid-blue and dashed-black curves, respectively. It is clear that τ_T increases, hence the concave region expands, as the buffer size is increased.

We estimate $\hat{f}_{\Theta_T}(\tau)$ for 1-10 parallel streams and three congestion control modules, namely, CUBIC, HTCP, and STCP. The overall variations of the estimated transition-RTT values w.r.t. number of parallel streams, buffer sizes, and TCP congestion control modules are shown in Fig. 6. For CUBIC with default buffer size, the transition-RTT τ_T increases from 0.4 ms for 1-3 parallel streams to 11.8 ms for 4 or more parallel streams. With normal buffer size, the transition-RTT τ_T remains consistently higher (at 45.6 ms, except for 2 steams) than with default buffer size, and further increases to 91.6 ms for 10 parallel streams. The τ_T estimate with the large buffer size is even larger than with both the default and normal buffer sizes; for example, 91.6 ms for 1-6 parallel streams (except 2 streams) and 183 ms for 7 or more parallel streams. Similar increasing trends of the estimate τ_T are also noted for HTCP and STCP, and thereby corroborate our inference that more streams and larger buffer sizes extend the concave region in addition to improving the throughput.



(a) CUBIC (b) HTCP (c) STCP

Fig. 6: Transition-RTT estimates with 1-10 streams and various buffer sizes for CUBIC, HTCP, and STCP.



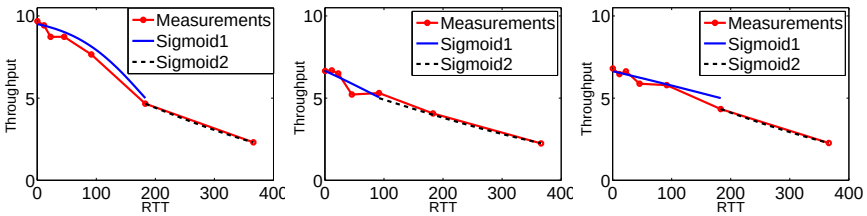
(a) XFS (b) direct I/O, 2 stripes (c) direct I/O, 8 stripes

Fig. 7: Mean throughput profiles of XFS and Lustre direct I/O file write transfer rates.

3.3 Estimates for File Transfers

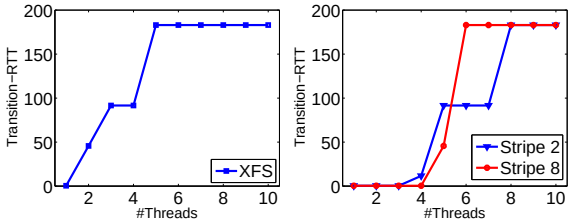
XDD High-performance disk-to-disk transfers between filesystems at different sites require the composition of complex file I/O and network subsystems, and host orchestration. For example, Lustre filesystem employs multiple OSTs to manage collections of disks, multiple OSSes to stripe file contents, and distributed MDSes 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 among them [22]. For Lustre filesystems, 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 I/O threads for read/write operations are specified at the transfer time. To sustain high throughput, I/O buffer size and the number of parallel threads are chosen to be sufficiently large, but this heuristic is not always optimal [18]. For instance, wide-area file transfers over 10 Gbps connections between two Lustre filesystems achieve transfer rates of only 1.5 Gbps, when striped across 8 storage servers, accessed with 8 MB buffers, and with 8 I/O and TCP threads [18], even though peak network memory-transfer rate and local file throughput are each close to 10 Gbps.

We measured file I/O and network throughput and file-transfer rates over Lustre and XFS filesystems for a suite of seven emulated connections in the 0–366 ms RTT range, which are used for memory transfer measurements in the previous section. We collected two sets of XDD disk-to-disk file transfer measurements, one from XFS to XFS and one from Lustre to Lustre, and each experiment is repeated 10 times. We considered both buffered I/O (the Linux default) and direct I/O options for Lustre. In the latter, XDD avoids the local copies of files on hosts by directly reading and writing into



(a) XFS (b) direct I/O, 2 stripes (c) direct I/O, 8 stripes

Fig. 8: Sigmoid regression fits of file transfer throughput profiles for XFS and Lustre.



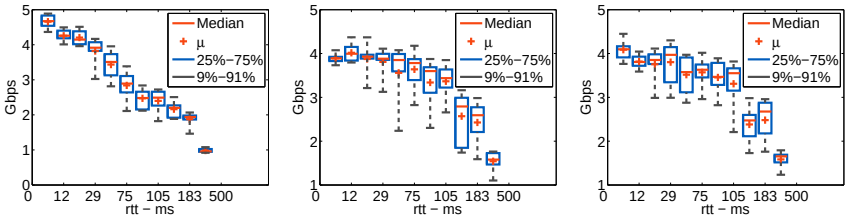
(a) XFS (b) Lustre direct I/O

Fig. 9: Transition-RTT estimates with respect to the number of flows for XFS and Lustre.

its buffers, which significantly improves the transfer rates. Results based on these measurements are summarized in [18]: (a) strategies of large buffers and higher parallelism do not always translate into higher transfer rates; (b) direct I/O 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 I/O streams, necessitate repeated measurements to ensure confidence in inferences based on them.

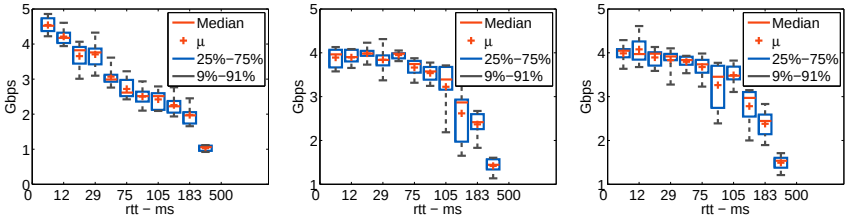
These file transfers are carried out using XDD, which spawns a set of *QThreads* to read a file from the source filesystem and subsequently transfer data over the network to the destination filesystem. We refer to each source-destination QThread connection as a *flow*. For XFS file transfers, the number of flows varies from 1 to 10, and the data transfer profile plot is shown in Fig. 7(a) for various RTTs. These profiles show monotonically increasing trends with respect to the number of flows. We use similar configuration for the Lustre experiments with the direct I/O option. In addition to varying the number of flows from 1 to 10, two different number of stripes were used: stripes 2 and 8, and the corresponding plots are respectively shown in Figs. 7(b) and 7(c). Similar to XFS profiles, the overall throughput exhibits predominantly increasing trends with respect to the number of flows; although compared to XFS, the Lustre throughput increases much slower with increasing flow counts in lower RTT cases. Comparing the performances of 2 vs. 8 striped Lustre, we notice that the use of 2 stripes yields somewhat higher transfer rates for lower flow counts. With more flows, overall throughput is higher, and 8 stripes is the better option.

To evaluate the transition-RTT values of file transfer profiles, we estimate the sigmoid fit $\hat{f}_{\Theta_E}(\tau)$ using measured XFS and Lustre throughput profiles. In Fig. 8, we



(a) cc=1, p=1 (b) cc=5, p=10 (c) cc=10, p=10

Fig. 10: GridFTP: Mean throughput profiles with CUBIC-Lustre-XFS configuration.



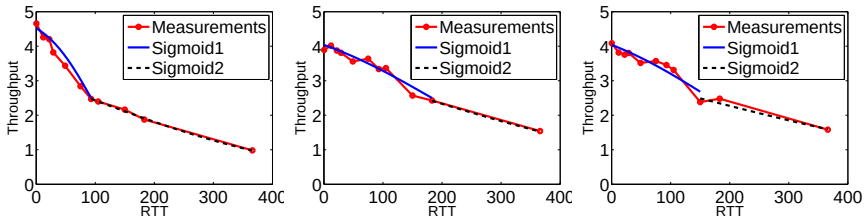
(a) cc=1, p=1 (b) cc=5, p=10 (c) cc=10, p=10

Fig. 11: GridFTP: Mean throughput profiles with HTCP-Lustre-XFS configuration.

demonstrate the fitted sigmoid plots overlayed on the mean profiles of measured throughput values with 6 parallel flows. We notice that all three variations of throughput profiles show concave and convex region to a different degree. For the XFS profile, the concave throughput profile at the smaller RTT values is more prominent than those in the Lustre profiles with both 2 and 8 stripes. In addition, we notice that transition-RTT values change depending on the file transfer throughput profiles; for example, both XFS and Lustre with 8 stripes show $\tau_T = 183$ ms with 6 flows, but Lustre with 2 stripes has $\tau_T = 91.6$ ms for this configuration.

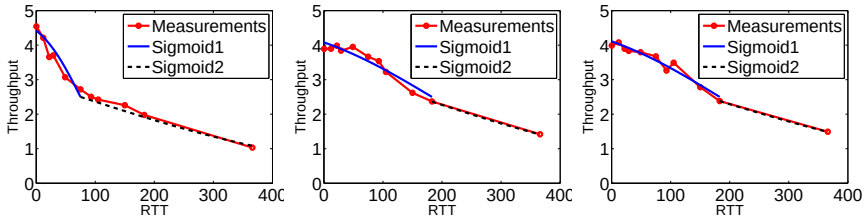
The overall characteristic of the transition-RTT values for XFS and Lustre filesystems at different number of flows is summarized in Fig. 9. For XFS, the transition-RTT values steadily increases from 0.4 ms for 1 parallel flows to 45.6 ms for 2 flows, 91.6 ms for 3-4 flows, and 183 ms for 5 or more flows. Similar increasing trends of the estimated τ_T values are also noted for Lustre filesystem in Fig. 9(b), but comparing them with Fig. 9(a) it becomes quite evident that XFS profiles have higher τ_T values, and hence wider concave profile regions, at smaller number of flows. Among the Lustre profiles with 2 and 8 stripes in Fig. 9(b), both produce only convex profiles for 4 or less number of flows. Then, transition-RTT values of Lustre with 8 stripes rise sharply to 183 ms in contrast to that with 2 stripes, indicating wider concave regions for 8-striped Lustre profile. For 8 flows or more, both 2-striped and 8-striped Lustre throughput profiles show the same τ_T values, but the measured throughput values are higher for 8-striped Lustre (see Figs. 7(b) and 7(c)).

GridFTP 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,



(a) cc=1, p=1 (b) cc=5, p=10 (c) cc=10, p=10

Fig. 12: Sigmoid fits of GridFTP throughput profiles with CUBIC-Lustre-XFS configuration.



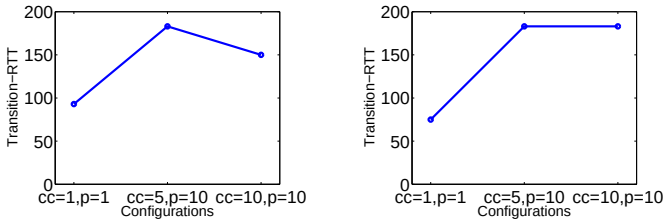
(a) cc=1, p=1 (b) cc=5, p=10 (c) cc=10, p=10

Fig. 13: Sigmoid fits of GridFTP throughput profiles with HTCP-Lustre-XFS configuration.

which provide support for striped transfers from multiple data sources. Data may be striped or interleaved across multiple servers, as in a parallel filesystem 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 I/O stream is handled by a single TCP stream, whereas such association is less strict in GridFTP.

Figs. 10 and 11 show GridFTP throughput performances for transfers between Lustre and XFS filesystems using CUBIC and HTCP congestion control modules. In each configuration, we vary the concurrency (cc) and parallelism (p) parameters, and collect throughput measurements across 11 different RTT values including the ones used in previous section. Here, XFS is mounted on SSD and provides throughput higher than 10 Gbps connection bandwidth. On the other hand, Lustre throughput is below 10Gbps and hence the transfer throughput is mainly limited by Lustre parameters. From these plots, it is quite evident that the overall throughput profiles of GridFTP are lower compared to XDD file transfer throughput values described in previous section.

To evaluate the transition-RTT values for GridFTP profiles, we estimate the sigmoid-based fit $\hat{f}_{\Theta_E}(\tau)$ of the measured GridFTP throughput values, and the corresponding results are shown in Figs. 12 and 13. We notice from Fig. 12(a) that, when the values of cc and p parameters are small, the concave throughput profiles extend only up to 93 ms. However, Figs. 12(b) and 12(c) show that, with the increase in cc and p values, the concave throughput regions of the GridFTP profiles become wider resulting



(a) CUBIC-Lustre-XFS (b) HTCP-Lustre-XFS

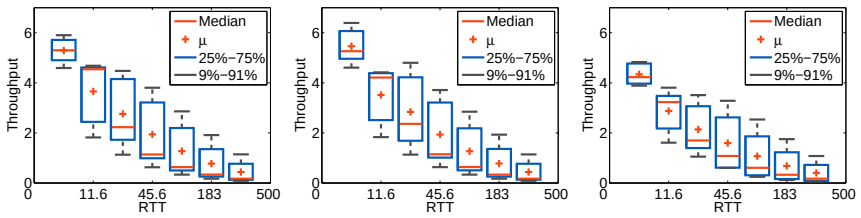
Fig. 14: Transition-RTT estimates with respect to the GridFTP configurations.

into $\tau_T = 183$ and 150 ms respectively at $cc = 5$ and 10 (along with $p = 10$). These variations of τ_T values with respect to cc and p parameters are plotted in Fig. 14(a) for the CUBIC-Lustre-XFS configuration of GridFTP profile. For the other GridFTP configuration, HTCP-Lustre-XFS, the sigmoid fitted models on the throughput profiles are shown in Fig. 13, and the resulting transition-RTT values are depicted in Fig. 14(b). Overall, they show similar trends as obtained with the CUBIC-Lustre-XFS configuration of GridFTP. The only notable difference is that, for the HTCP-Lustre-XFS configuration of GridFTP, the τ_T values stays at 183 ms for both $cc = 5$ and 10 (when $p = 10$), indicating monotonic trend with cc and p parameters.

Lustre Over LNet As mentioned earlier, Lustre filesystem employs multiple OSTs to manage collections of disks, and multiple 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 filesystem 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 filesystem is augmented with Ethernet Lustre clients, and LNet routers are utilized to make IB-based OSSs available over wide-area Ethernet connections [16]. Compared to GridFTP and XDD, which are software applications, the implementation of LNet routers requires more changes to the infrastructure.

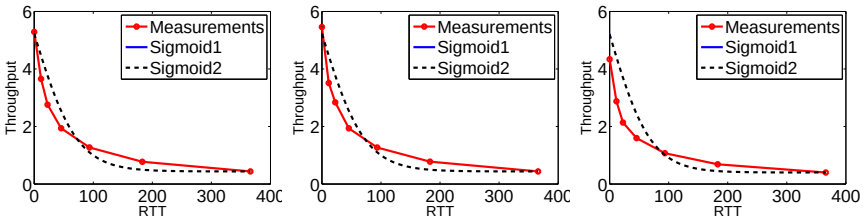
The throughput profiles of Lustre over wide-area connections using LNet routers are shown in Fig. 15. For these measurements, we use two classes of hosts: (i) `bohr05` (b5) and `bohr06` (b6) are DTN servers with 48 cores, and (ii) `tait1` (t1) and `tait2` (t2) are compute nodes of a cluster with 32 cores. In each case, one of the node is used as a LNet router which extends IB network to wide-area TCP/IP connection. As shown in Section 3.1, TCP buffer size could have a significant impact on the concave-convex shape of the profile.

In Fig. 16, we demonstrate the sigmoid fit $\hat{f}_{\Theta_E}(\tau)$ along with the measured throughput profiles for LNet router configurations. As noted from Fig. 15, the profiles are entirely convex in all our LNet configurations, and consequently we get only the convex portions of the sigmoid fits. Therefore, the transition-RTT values for all LNet experiments are estimated to 0.1 ms, as shown in Fig. 17. We also used two LNet buffer sizes, 50M and 2G, which corresponded to convex and concave profiles for the under-



(a) b5-b6-2G-tuned (b) b5-b6-50M-tuned (c) t2-t1-2G-tuned

Fig. 15: LNet: Mean throughput profiles.



(a) b5-b6-2G-tuned (b) b5-b6-50M-tuned (c) t2-t1-2G-tuned

Fig. 16: Sigmoid regression fits of LNet throughput profiles.

lying TCP throughput profiles, as illustrated in Fig. 4. However, for file transfers via LNet routers, in all the three configurations, the estimated throughput profile $\hat{f}_{\Theta_E}(\tau)$ as a function of RTT is entirely convex, which indicates that the source of convexity is elsewhere. Indeed, the LNet credits limit the number of packets in transit over the Lustre filesystem, which in turn limits the number of packets in transit over TCP connection. Such limit is equivalent to TCP buffer limit, which explains the convex profile indicated by our estimate $\hat{f}_{\Theta_E}(\tau)$.

4 Confidence Probability Estimates

In the previous section, the convex-concave estimate $\hat{f}_{\Theta_A}(\tau)$ based on measurements has been important in assessing the effectiveness of the transport methods and configurations. Now, we address its soundness from a finite sample statistics perspective [24], particularly in assessing properties of $\Theta_A(\tau)$ that depend on (potentially) infinite-dimensional $\mathbf{P}_{\Theta_A(\tau)}$. We consider the *profile regression* given by

$$\bar{\Theta}_A(\tau) = E\left[\Theta_A(\tau)\right] = \int \Theta_A(\tau) d\mathbf{P}_{\Theta_A(\tau)},$$

which is approximated by $\hat{f}_{\Theta_A}(\tau)$ using *finite* independent and identically distributed (i.i.d.) measurements $\theta(\tau_k, t_j^k)$ at τ_k, $k = 1, 2, \ldots, n$, collected at times t_j^k, $j = 1, 2, \ldots, n_k$. The critical concave-convex property provided by $\hat{f}_{\Theta_A}(\tau)$ is only an approximation of the true concave-convex property of $\bar{\Theta}_A(\tau)$, which depends on the complex distribution $\mathbf{P}_{\Theta_A(\tau)}$. In general, it is not practical to estimate $\mathbf{P}_{\Theta_A(\tau)}$ since it depends on hosts, filesystems, and network connections, as well as on software components, including TCP congestion control kernel modules, and file- and memory-transfer application modules. For

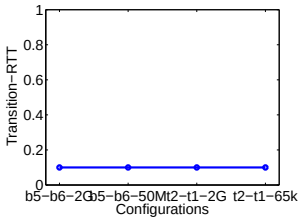


Fig. 17: Transition-RTT estimates with respect to various LNet setups.

the estimate $\hat{f}_{\Theta_A}(\tau)$ computed based solely on measurements, we derive probability that its expected error is close to the optimal error, independent of $\mathbf{P}_{\Theta_A(\tau)}$.

Consider an estimate $f(.)$ of $\bar{\Theta}_A(.)$ chosen from a function class $\mathscr{M}$. The *expected error* $I(f)$ of the estimator f is

$$I(f) = \int [f(\tau) - \theta(\tau,t)]^2 d\mathbf{P}_{\theta(\tau,t)}.$$

The *best estimator* f^* is given by $I(f^*) = \min_{f\in\mathscr{M}} I(f)$, which in general is not possible obtain since $\mathbf{P}_{\theta(\tau,t)}$ in unknown. The *empirical error* of estimator f based on measurements is given by

$$\hat{I}(f) = \frac{1}{n_T}\sum_{k=1}^{n}\sum_{j=1}^{n_k}[f(\tau_k) - \theta(\tau_k,t_j)]^2,$$

which is a scaled version of the sum-squared error $\hat{S}(f)$ in Eq. (3.1), that is $\hat{I}(f) = \frac{1}{n_T}\hat{S}(f)$, where $n_T = \sum_{k=1}^{n} n_k$. The *best empirical estimator* $\hat{f} \in \mathscr{M}$ minimizes the empirical error, that is, $\hat{I}(\hat{f}) = \min_{f\in\mathscr{M}} \hat{I}(f)$. Thus, $\hat{f}$ is estimated based entirely on measurements as an approximation to f^*.

Let us consider a class of flipped sigmoid functions with bounded weights $\mathscr{M}_{a,\tau} = \{g_{a,\tau} : a \in [-A,A], \tau \in [-T,T]\}$. The estimator $\hat{f}_{\Theta_A}$ is chosen from the set of estimators $\mathscr{M}_{\Theta_A} = \left\{f_{\Theta_A} : g_{a_1,\tau_1}, g_{a_2,\tau_2} \in \mathscr{M}_{a,\tau}\right\}$, which is not guaranteed to minimize the empirical error since τ_T is limited to τ_i. Then, the empirical error of our estimator $\hat{f}_{\Theta_A}$ is given by $\hat{I}\left(\hat{f}_{\Theta_A}\right) = \hat{I}(\hat{f}) + \hat{\varepsilon}$ in terms of the empirical best estimator $\hat{f}$, where $\hat{\varepsilon} \geq 0$. We now show that the expected error of $\hat{f}_{\Theta_A}$ is within $\varepsilon + \hat{\varepsilon}$ of optimal error $I(f^*)$, for any $\varepsilon > 0$, with a probability that improves with the number of measurements and is independent of the complex, unknown $\mathbf{P}_{\theta(\tau,t)}$.

Theorem 1. *For $\hat{f}_{\Theta_A}$ estimated based on $n_T = \sum_{k=1}^{n} n_k$ i.i.d. measurements, the probability that its expected error is within $\varepsilon + \hat{\varepsilon}$ of the optimal error is upper bounded as*

$$\mathbf{P}\left\{I\left(\hat{f}_{\Theta_A}\right) - I(f^*) > \varepsilon + \hat{\varepsilon}\right\} \;\leq\; K\left(\frac{AT}{\varepsilon^2}\right)e^{-\varepsilon^2 n_T/512},$$

for any $\varepsilon > 0$ and $K = 2048$.

Proof. We first note that

$$\mathbf{P}\left\{I\left(f_{\Theta_A}\right)-I(f^*)>\varepsilon+\hat{\varepsilon}\right\}\leq\mathbf{P}\left\{I\left(g_{a_1,\tau_{a_1}}\right)+I\left(g_{a_2,\tau_{a_2}}\right)-I(f^*)>\varepsilon+\hat{\varepsilon}\right\}$$

since $I\left(g_{a_1,\tau_{a_1}}\right)+I\left(g_{a_2,\tau_{a_2}}\right)\geq I\left(f_{\Theta_A}\right)$ wherein both $g_{a_1,\tau_{a_1}}$ and $g_{a_2,\tau_{a_2}}$ are expanded to the entire range of τ. Then, we can write

$$\begin{aligned}\mathbf{P}\left\{I\left(g_{a_1,\tau_{a_1}}\right)+I\left(g_{a_2,\tau_{a_2}}\right)-I(f^*)>\varepsilon+\hat{\varepsilon}\right\}\leq\mathbf{P}&\left\{I\left(g_{a_1,\tau_{a_1}}\right)-I(g^*_{a,\tau_a})>\varepsilon/2+\hat{\varepsilon}/2\right\}\\&+\mathbf{P}\left\{I\left(g_{a_2,\tau_{a_2}}\right)-I(g^*_{a,\tau_a})>\varepsilon/2+\hat{\varepsilon}/2\right\},\end{aligned}$$

where $I(g^*_{a,\tau_a})=\min\limits_{g_{a,\tau_a}\in\mathscr{M}_{a,\tau}}I(g_{a,\tau_a})$. This inequality follows by noting that the condition

$$I\left(g_{a_i,\tau_{a_i}}\right)-I(g^*_{a,\tau_a})<\varepsilon/2+\hat{\varepsilon}/2$$

for both $i=1,2$ implies that $I\left(g_{a_1,\tau_{a_1}}\right)+I\left(g_{a_2,\tau_{a_2}}\right)-I(f^*)<\varepsilon+\hat{\varepsilon}$. Equivalently, the opposite of the latter condition implies that of the former, and thus the probability of the former is upper bounded by that of the latter, establishing the above inequality.

Next, by using Vapnik-Chervonenkis theory [24], we have

$$\begin{aligned}&\mathbf{P}\left\{I\left(g_{a_1,\tau_{a_1}}\right)-I(g^*_{a,\tau_a})>\varepsilon/2+\hat{\varepsilon}/2\right\}\\&\leq\mathbf{P}\left\{\max_{h\in\mathscr{M}_{a,\tau}}|I(h)-\hat{I}(h)|>\varepsilon/4\right\}\leq 8\mathscr{N}_\infty(\varepsilon/32,\mathscr{M}_{a,\tau})\,e^{-\varepsilon^2 n_T/512}\end{aligned}$$

where $\theta(\tau,t)\leq 1$, and $\mathscr{N}_\infty(\varepsilon,\mathscr{M}_{a,\tau})$ is the ε-cover of $\mathscr{M}_{a,\tau}$ under L_∞ norm (see [7, 12] for the last bound).

We now show that $g_{a,\tau_a}\in\mathscr{M}_{a,\tau}$ is Lipschitz as follows: (i) $|g_{a,\tau}(x)-g_{a\prime,\tau}(x)|<AT/4\varepsilon$ for all x under the condition $|a-a\prime|\leq\varepsilon$, and (ii) $|g_{a,\tau}(x)-g_{a,\tau\prime}(x)|<AT/4\varepsilon$ for all x under the condition $|\tau-\tau\prime|\leq\varepsilon$. The Lipschitz constant in (i) is estimated by the maximum derivative with respect to a (part (ii) is similar). Let $z=a(\tau-\tau_a)$ and $\sigma(z)=1/(1+e^{-z})$, such that $g_{a,\tau_a}(\tau)=1-\sigma(z)$. Then, we have
$\frac{d\sigma(z)}{dz}=a\sigma(z)[1-\sigma(z)]\leq a/4$. Thus, we get [13]

$$\frac{dg_{a,\tau_a}}{da}=\frac{d\sigma(z)}{dz}\tau_a\leq AT/4\,.$$

We consider a two-dimensional grid $[-A,A]\times[-T,T]$ with $4AT/\varepsilon^2$ points equally spaced in each dimension. Then, for any (a,τ_a), there exists a grid point (b,τ_b) such that $|a-b|\leq\varepsilon$ and $|\tau_a-\tau_b|\leq\varepsilon$, which in turn assures that $\|\,g_{a,\tau_a}-g_{b,\tau_b}\,\|_\infty\leq AT/4\varepsilon$. Consequently, we have

$$\mathscr{N}_\infty(\varepsilon/32,\mathscr{M}_{a,\tau})\leq 128\left(\frac{AT}{\varepsilon^2}\right).$$

By applying this bound for both $g_{a_1,\tau_{a_1}}$ and $g_{a_2,\tau_{a_2}}$, we obtain

$$\mathbf{P}\left\{I\left(\hat{f}_{\Theta_A}\right)-I(f^*)>\varepsilon+\hat{\varepsilon}\right\}<16\mathscr{N}_\infty(\varepsilon/32,\mathscr{M}_{a,\tau})\,e^{-\varepsilon^2 n_T/512},$$

which establishes the probability bound. □

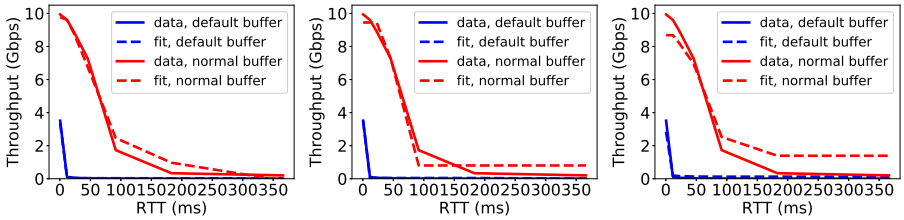
This result ensures that $I\left(\hat{f}_{\Theta_A}\right) - I(f^*) < \varepsilon + \hat{\varepsilon}$ with a probability at least $\alpha = \left[1 - K\left(\frac{AT}{\varepsilon^2}\right) e^{-\varepsilon^2 n_T/512}\right]$, which approaches to 1 as $n_T \to \infty$. In particular, the exponential term decays faster in n_T than other terms, and hence for a sufficiently large n_T a given confidence probability α can be assured. This performance guarantee is independent of how complex the underlying distribution $\mathbf{P}_{\Theta_A(\tau)}$ is, and thus provides confidence in the inferences based on the concavity-convexity properties of the estimate $\hat{f}_{\Theta_A}$ derived entirely from measurements. Similar performance guarantees have been provided for a somewhat different problem of network throughput profile estimation in [14, 17]. It is interesting to note that the exponent of ε in the bound for the cover size $\mathscr{N}_\infty(\varepsilon/32, \mathscr{M}_{a,\tau})$ is 2 in the above proof, although 5 parameters ($a_1, a_2, \tau_{a_1}, \tau_{a_2}$ and τ_T) are needed to estimate $\hat{f}_{\Theta_A}$. In this sense, the underlying structure of $\hat{f}_{\Theta_A}$ reduces the estimation dimensionality from 5 to 2. For a similar cover size estimate for feedforward sigmoidal neural networks, for example, the exponent of ε would be the total number of neural network parameters (much larger than 2).

5 Generic Machine Learning Methods

To gain further insights into throughput performance, we apply generic machine learning modules, available in Python and MATLAB, for obtaining regression-based estimates of throughput profiles. The regression-fits in general not only indicate the overall throughput trends as functions of various parameters, but also provide throughput predictions for configurations at which measurements have not been collected.

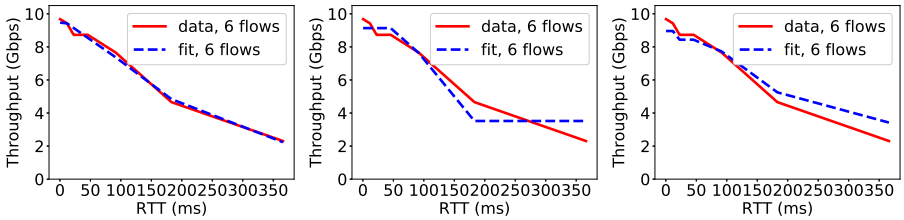
From the Python-based machine learning tools, we use three well-known learning approaches: (i) artificial neural network (ANN), (ii) random forest (bagging), and (iii) gradient boosting. From MATALB, we use (i) support vector machines (SVM) and (ii) Gaussian kernel estimates, for GridFTP measurements. We implement ANN in the `tensorflow` framework with the following specifications: number of hidden layers = 2, number of hidden units = 8 and 5, learning rate = 0.001, ReLU activation units, and Adam optimizer. Both the weights and bias terms of the ANN are initialized with normal random variables. The random forest model is also developed in the `tensorflow` framework, specifically with the *tensor_forest* module. In this setup, we use number of trees = 2 and number of maximum nodes = 10, and execute the tensor_forest graph in the regression mode. The gradient boosting method is implemented using the `scikit-learn` library, along with the following parameters: number of trees = 1500, maximum depth = 2, learning rate = 0.001, and minimum samples per split = 10. Furthermore, in the all the three learning methods, we use mean-squared error as the cost metric for optimization purposes.

Memory Fig. 18 shows the fitted regression curves obtained via ANN, random forest, and gradient boosting methods, when they are applied to the CUBIC TCP throughput measurements. Specifically, in Fig. 18, we show only the cases with single flow using the default and normal buffer sizes. The mean throughput profiles of the measured data are shown with bold lines, whereas the corresponding fitted models are shown via dotted lines. The corresponding sigmoid-function based regression fits are shown in Figs. 5(a)



(a) artificial neural net (b) random forest (c) gradient boosting

Fig. 18: CUBIC: Generic regression fits to default and normal buffer measurements with 1 flow.

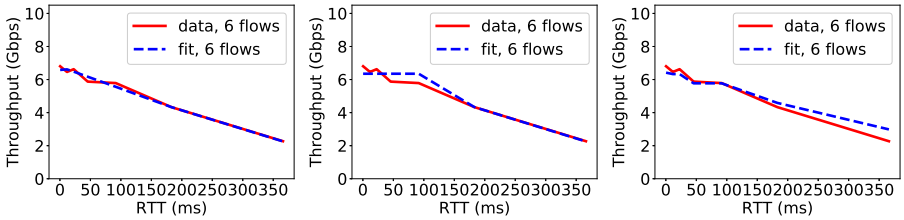


(a) artificial neural net (b) random forest (c) gradient boosting

Fig. 19: XDD XFS: Generic regression fits to file transfer measurements at 6 flows.

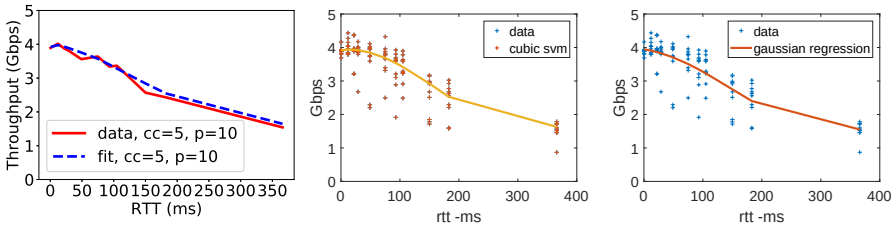
and 5(b). At the chosen parameter setting for the generic learning methods, we notice from Fig. 18 that the ANN and random forest based regression models show better fits to these CUBIC TCP datasets than the gradient boosting based model. Particularly, the learning models show better accuracy (i.e., smaller error) with the default-buffer configuration and at somewhat small RTT values. However, from these generic regression fits, the concave-convex portions of throughput profiles and the associated transition-RTT values are not quite as clear as with sigmoid regression fits.

XDD For the disk-to-disk file transfer throughput measurements, we also apply the ANN, random forest, and gradient boosting learning methods to obtain the regression-fit estimates. In Figs. 19 and 20, we depict the performances of these learning methods respectively for the XFS and Lustre (8-striped with direct I/O option) filesystems. To compare with the corresponding sigmoid-function based regression method (see Figs. 8(a) and 8(c)), we only show the fitted lines at 6 parallel flows. The parameters of the learning methods are kept the same as those used for the TCP throughput measurements. In general, ANN model show the best accuracies, followed by the random forest and gradient boosting models, for both XFS and Lustre direct I/O throughput measurements. The fitted lines via random forest and gradient boosting methods show large errors from the XFS measurements particularly at larger RTT values. Comparing the performances between the XFS and Lustre throughputs, we notice that the fitting accuracy is little better with the Lustre datasets than XFS. However, from the concavity-convexity perspective, these generic fits do not provide as clear information as obtained via the sigmoid based regression approach.



(a) artificial neural net (b) random forest (c) gradient boosting

Fig. 20: XDD Lustre: Generic regression fits to 8-striped direct I/O measurements at 6 flows.

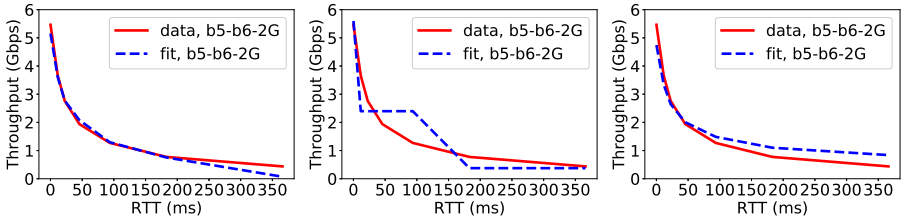


(a) artificial neural net (b) support vector machine (c) Gaussian kernel

Fig. 21: GridFTP: Generic regression fits to CUBIC-Lustre-XFS file transfer measurements.

GridFTP The generic regression model fits to the GridFTP measurements are demonstrated in Fig. 21 for the CUBIC-Lustre-XFS configuration with the parameter values $cc = 5$ and $p = 10$, using ANN model from Python, and SVM and Gaussian kernel models from MATLAB. These plots are equivalent to Figs. 12(b) depicting the sigmoid regression fit. Overall, the characteristics of all the fitted models seems quite similar to each other. In particular, from the SVM fit in Fig. 21(b), we can observe the concave and convex portions of the fitted model respectively at the smaller and larger RTT values. However, exact specification of the transition-RTT value requires one more post-processing step on the fitted SVM model, which was not required in the sigmoid-regression method. The concave portion is also present in Gaussian kernel fit although less prominent compared to SVM fit, as shown in Fig. 21(c). Comparatively, the concave-convex shape is less discernable in neural network fit in Fig. 21(a), whereas the transition is apparent in the other two fits.

LNet The regression fits obtained using ANN, random forest, and gradient boosting techniques to the LNet throughput measurements are depicted in Fig. 22 for the b5-b6-2G configuration. Fig. 16(a) depicts the equivalent sigmoid-regression fit. We do not include the generic regression fits to the other LNet datasets as they have similar trends. Among the three generic learning methods, the ANN model shows the best accuracy in fitting the measured LNet throughput data, followed by the gradient boosting and random forest models. Overall, these regression fits capture the entirely-convex trends of LNet throughput profiles, except the random forest fit, which is somewhat non-representative of convex profile.



(a) artificial neural net (b) random forest (c) gradient boosting

Fig. 22: LNet: Generic regression fits to b5-b6-2G file transfer measurements.

6 Conclusions

We presented a learning approach that explicitly captures the dichotomous throughput profiles observed in throughput measurements collected in a number of data transport scenarios. It identifies concave regions at lower RTTs that indicate near-optimal performance, and convex regions at higher RTTs that indicate bottlenecks due to factors such as buffer or credit limits. This approach uses sigmoid functions to characterize the concave and convex regions in throughput profiles, and also provides distribution-free confidence estimates for the closeness of its expected error to optimal error. We applied this method to throughput measurements of memory and file transfers over connections ranging from local to cross-country to round-the-earth distances. The convex-concave profile estimates enabled us to infer the near optimality of transfers in practical scenarios, such as XDD transfers between XFS filesystems, based on concave regions. In addition, this approach also enabled us to infer performance limits, such as in case of wide-area Lustre extensions using LNet routers, based on convex regions. A direct application of generic machine learning packages did not always highlight these critical performance regions nor provided as sharp confidence estimates, thereby making a case for customized machine learning methods such as the one proposed here.

In terms of future directions, characterizations of latency, jitter, and other dynamic properties will be of interest for data transport infrastructures, in addition to throughput profiles considered here. Additionally, it would be of future interest to derive confidence estimates for the transition-RTT computed using the measurements. Extensions of the proposed estimates and analyses to data transport over shared network connections would also be of future interest.

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 *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, SC '05, pages 54–64, Washington, DC, 2005. IEEE Computer Society.
2. Bryce Allen, John Bresnahan, Lisa Childers, Ian Foster, Gopi Kandaswamy, Raj Kettimuthu, Jack Kordas, Mike Link, Stuart Martin, Karl Pickett, and Steven Tuecke. Software as a service for data scientists. *Communications of the ACM*, 55(2):81–88, February 2012.

3. Aspera high-speed file transfer. http://www-03.ibm.com/software/products/en/high-speed-file-transfer.
4. N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson. BBR: Congestion based congestion control. *ACM Qeueue*, 14(5), Dec. 2016.
5. General Parallel File System, https://www.ibm.com/support/knowledgecenter/en/SSFKCN/gpfs_welcome.html.
6. Y. Gu and R. L. Grossman. UDT: UDP-based data transfer for high-speed wide area networks. *Computer Networks*, 51(7), 2007.
7. A. Krzyzak, T. Linder, and G. Lugosi. Nonparametric estimation and classification using radial basis function nets and empirical risk minimization. *IEEE Transactions on Neural Networks*, 7(2):475–487, 1996.
8. Q. Liu, N. S. V. Rao, C. Q. Wu, D. Yun, R. Kettimuthu, and I. Foster. Measurement-based performance profiles and dynamics of udt over dedicated connections. In *International Conference on Network Protocols*. Singapore, Nov. 2016.
9. Z. Liu, R. Kettimuthu, I. Foster, and N. S. V. Rao. Cross-geography scientific data transferring trends and behavior. In *ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC)*. June 2018.
10. Lustre Basics, https://www.olcf.ornl.gov/kb_articles/lustre-basics.
11. Multi-core aware data transfer middleware. ndtm.fnal.gov.
12. D. Pollard. *Convergence of Stochastic Processes*. Springer-Verlag, New York, 1984.
13. N. S. V. Rao. Simple sample bound for feedforward sigmoid networks with bounded weights. *Neurocomputing*, 29:115–122, 1999.
14. N. S. V. Rao. Overlay networks of in-situ instruments for probabilistic guarantees on message delays in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 22(1), 2004.
15. N. S. V. Rao. Finite-sample generalization theory for machine learning practice for science. In *DOE ASCR Scientific Machine Learning Workshop*, 2018.
16. 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.
17. 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 (HPDC)*, Washington, DC, July-August. 2017.
18. 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.
19. 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 *The 26th International Conference on Computer Communications and Network(ICCCN 2017)*, 2017.
20. I. Rhee and L. Xu. CUBIC: A new TCP-friendly high-speed TCP variant. In *Proceedings of the Third International Workshop on Protocols for Fast Long-Distance Networks*, 2005.
21. 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.
22. 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 (MSST)*, pages 1–6, May 2011.

23. R. N. Shorten and D. J. Leith. H-TCP: TCP for high-speed and long-distance networks. In *Proceedings of the Third International Workshop on Protocols for Fast Long-Distance Networks*, 2004.
24. V. N. Vapnik. *Statistical Learning Theory*. John-Wiley and Sons, New York, 1998.
25. XDD - The eXtreme dd toolset, https://github.com/bws/xdd.
26. XFS, http://xfs.org.