Measurements of File Transfer Rates Over Dedicated Long-Haul Connections

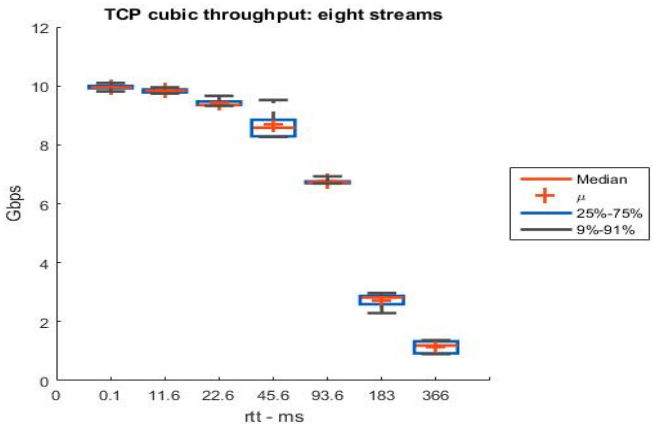
Nageswara S. V. Rao*, Greg Hinkel*, Neena Imam*, Bradley W. Settlemyer†
*Oak Ridge National Laboratory, †Los Alamos National Laboratory

***Abstract*—Wide-area file transfers are an integral part of several High-Performance Computing (HPC) scenarios. Dedicated network connections with high capacity, low loss rate and low competing traffic, are increasingly being provisioned over current HPC infrastructures to support such transfers. To gain insights into these file transfers, we collected transfer rate measurements for Lustre and xfs file systems between dedicated multi-core servers over emulated 10 Gbps connections with round trip times (rtt) in 0-366 ms range. Memory transfer throughput over these connections is measured using iperf, and file IO throughput on host systems is measured using xddprof. We consider two file system configurations: Lustre over IB network and xfs over SSD connected to PCI bus. Files are transferred using xdd across these connections, and the transfer rates are measured, which indicate the need to jointly optimize the connection and host file IO parameters to achieve peak transfer rates. In particular, these measurements indicate that (i) peak file transfer rate is lower than peak connection and host IO throughput, in some cases by as much as 50% or lower, (ii) xdd request sizes that achieve peak throughput for host file IO do not necessarily lead to peak file transfer rates, and (iii) parallelism in host IO and TCP transport does not always improve the file transfer rates.**

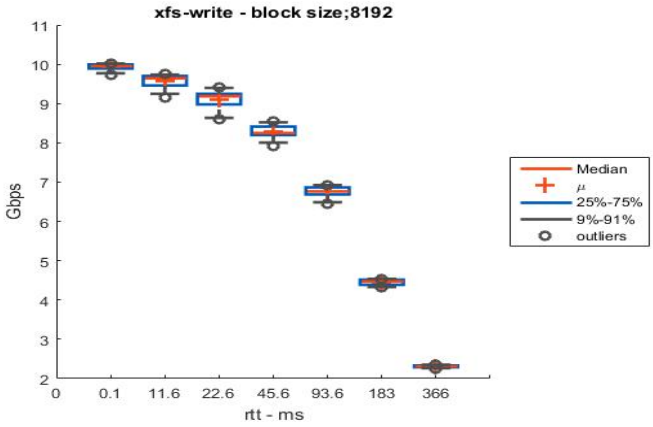
Keywords: File sytems, TCP, long-haul connections, throughput.

I. Introduction

In several High-Performance Computing (HPC) scenarios, the workflows require wide-area data transfers over high capacity dedicated networks. Such transfers often involve file transfers between supercomputer and storage sites that are connected over long-haul networks. To support these transfers, network infrastructures, such as Department of Energy's (DOE) ESnet, are being enhanced to provide on-demand, dedicated connections [1] with very low losses and limited or no competing traffic. Also, file systems such as Lustre are being appropriately scaled up with disk complexes served by multiple Object Storage Targets (OST) and Object Storage Servers (OSS), and distributed Meta Data Servers (MDS)[3]. Furthermore, dedicated hosts such as the Data Transfer Nodes (DTN) [2] of DOE, are being equipped with multiple cores some of which can be dedicated for network tasks while others perform file IO operations. File transfers in these scenarios represent a convergence of data transfer capabilities that have been traditionally carried out over short distances using InfiniBand (IB) and those over long-haul connections using Transmission Control Protocol (TCP). In view of long distances between the transfer sites, TCP is a natural candidate for such data transfers. However, sustaining high file transfer rates requires jointly optimizing file IO and TCP parameters to match end systems and connection [5].



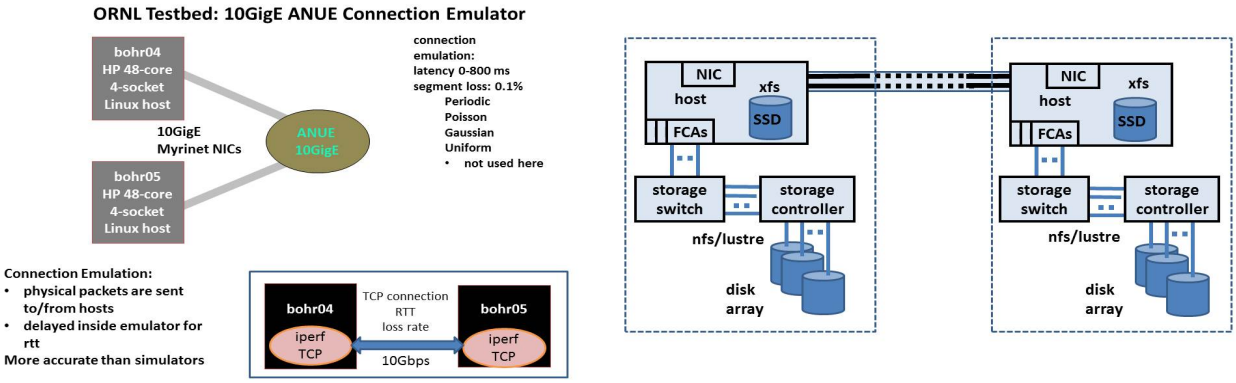
(a) CUBIC TCP throughput measured using iperf



(b) xfs write transfer rate measured by xdd with 8 streams

Fig. 1. TCP throughput and xfs file transfer rate over WAN connections with rtt in 0-366 ms range.

To gain insights into optimizations needed for these transfers, we systematically collected file IO and TCP throughput measurements and also file transfer rates over dedicated connections emulated in hardware for a wide range of round trip times (rtt). Measurements in a simple illustrative scenario are shown in Figure 1, where xfs file systems are mounted on Solid State Drives (SSD) connected over PCI bus at end hosts. The peak file IO throughput measured using xddprof [7] is around 40Gbps, and is above 10Gbps capacity of the network connection. TCP throughput with 8 parallel streams



(a) emulated long-haul connections

(b) host configurations

Fig. 2. Testbed configurations of emulated long-haul connections and host systems with Lustre and xfs file systems.

over 10GigE dedicated connection measured using iperf [4] is close to the peak at smaller rtt but is lower for longer rtt as shown in Figure 1(a); TCP parameters are set to the recommended values for 200ms rtt [6]. The file transfer rate for write IO operation using xdd with 8 threads between these hosts is shown in Figure 1(b); XDD is a file transfer tool between disk systems with a wide set of tunable parameters [7]. For small rtt, the file transfer rate is within 10% below iperf TCP measurements and becomes comparable at larger rtts, but it exhibits somewhat higher statistical variations. In several other cases, however, the gap between the two is much wider as will be described subsequently in this paper.

In general, compared to file IO and TCP throughput, the file transfer rates showed much more complex variations both statistically and with respect to XDD parameters of request or block size and the number of threads or parallel streams. We present file transfer rate measurements for various XDD parameters with an objective of gaining overall qualitative insights into this class of file transfers. For this purpose, we collect measurements at several parameter settings some of which are non-optimal and may be further refined to improve the transfer rates.

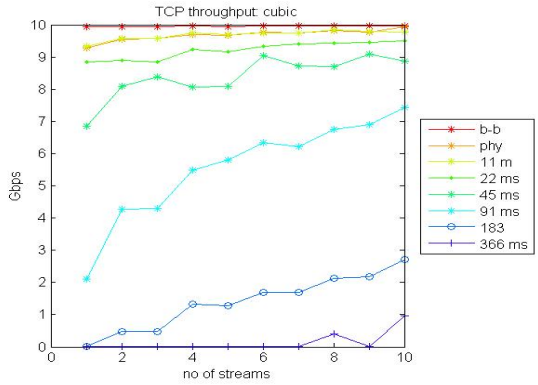
The measurements of file transfer rates are collected over ORNL testbed for Lustre file system mounted over IB network and xfs system mounted on SSD connected over PCI bus. The file transfers are carried out across these connections using XDD and the transfer rates are measured. These rates are determined by the file system, data transfer hosts and network connection, and equally importantly, by the interactions between them. Overall, these measurements indicate the need to jointly optimize the parameters of these constituent systems to achieve peak transfer rates, since their individual optimizations do not necessarily lead to peak end-to-end file transfer rates. In particular, an analysis of these measurements indicates that (i) peak file transfer rate is lower than peak connection and host IO throughput, in some cases by as much as 50% or lower, (ii) request sizes that achieve peak throughput for host file IO do not necessarily lead to peak file transfer rate, and (iii) increased parallelism in host IO and TCP transport does not always improve the file transfer rates. In this paper, we provide a brief description of the measurements and a summary of these analyses.

This paper is organized as follows. We describe our experimental setup in Section II. TCP throughput and memory transfer rate measurements are described in Section III. File IO throughput measurements on end host systems are briefly described in Section IV. Measurements of file transfer rates and their analyses are presented in Section V, wherein Lustre and xfs files systems are discussed in Sections V-A and V-B, respectively. Overall summary of our results and some future research directions are presented in Section VI.

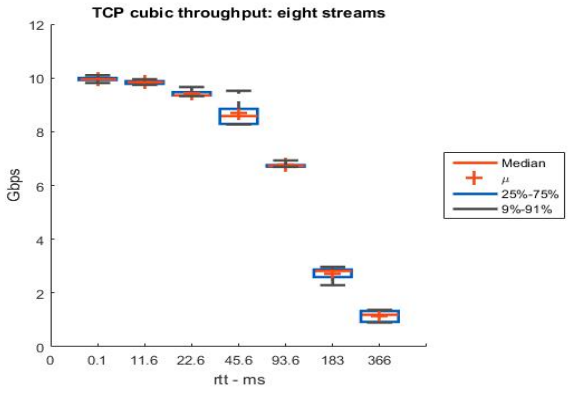
II. Experimental Setup

We collected measurements of file transfer rates over Lustre and xfs file systems between two dedicated 48-core Linux servers over emulated 10 Gbps connections for rtt τ =11.6, 22.6. 45.6, 91.5, 183 and 366 ms. The connections are emulated using ANUE-ixia devices to which the host 10GigE interfaces are connected as shown in Figure 2(a). The lower rtts represents US cross-country connections, for example, ones between two DOE sites, and higher rtts represent trans-continental connections.

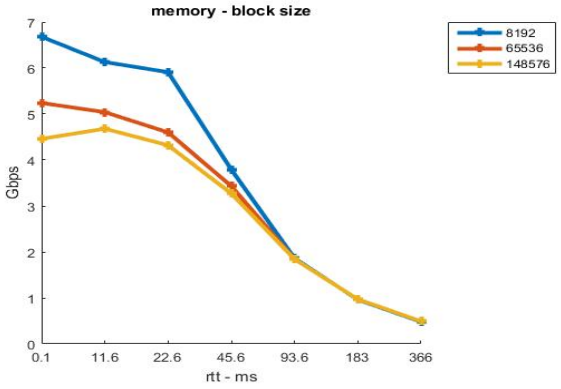
The file system configurations are shown in Figure 2(b), where Lustre file system is mounted over local IB network. xfs file system is mounted locally on each host over SSDs connected to its PCI bus. We utilized XDD [7] for transferring files between the hosts, which employs multiple IO threads for reading and writing files to disks, and TCP for network transport; we also measured the file transfer rates. It utilizes the same threads for file IO and TCP transport, and thus the number of IO threads is the same as the number of parallel TCP streams. We collected TCP throughput measurements that correspond to memory transfer rates over these connections using iperf and XDD for single and multiple streams; here we used CUBIC TCP congestion control module which is default on Linux systems. File IO throughput at end systems



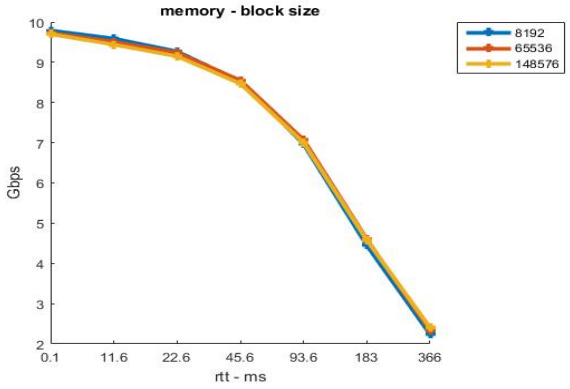
(a) TCP throughput for multiple streams



(b) TCP throughput for 8 streams



(c) memory transfer rate for 1 stream



(d) memory transfer rate for 8 streams

Fig. 3. Average TCP throughput and xdd memory transfer rate measurements between two 48-core host systems over dedicated 10gigE connections.

is measured using XDDprof that sweeps file IO parameters including the number of IO threads and request size, which we consider here.

III. TCP Throughput and Memory Transfer Rates

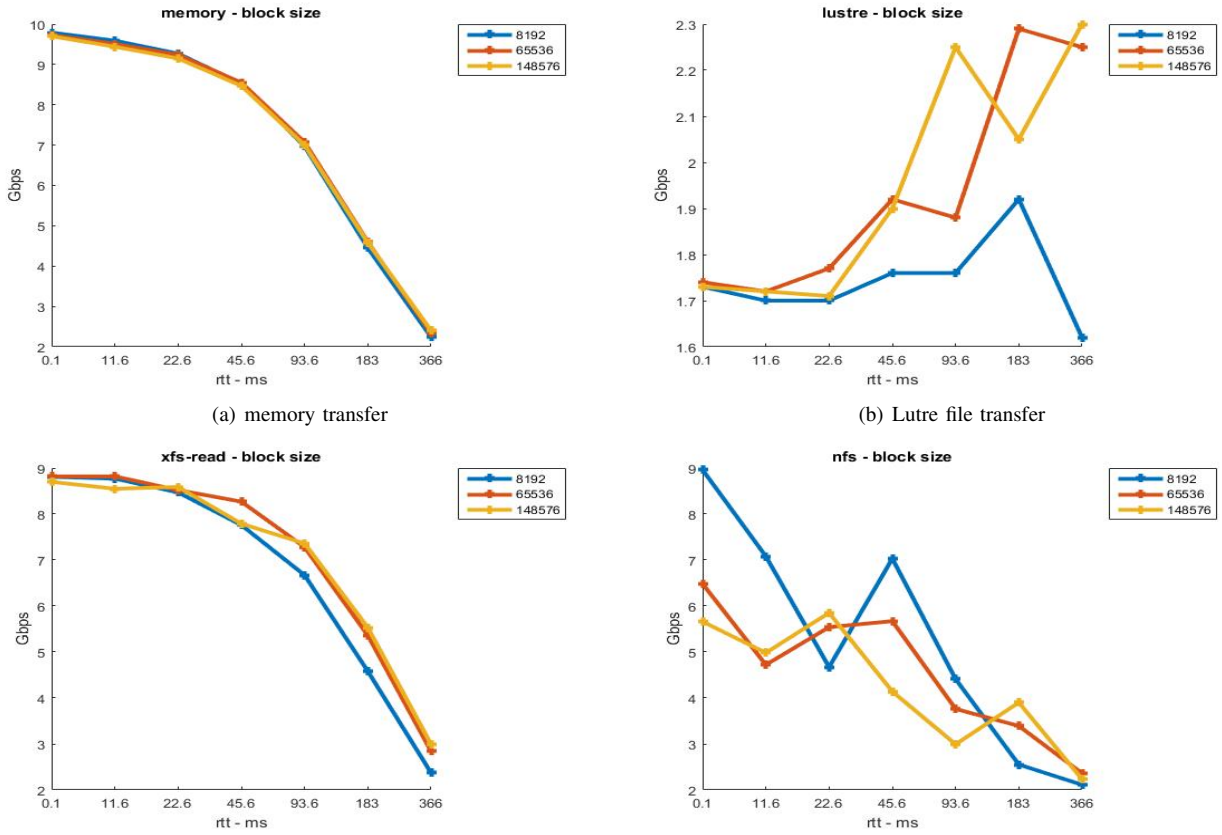
TCP throughput measurements for memory-to-memory transfers are collected using iperf-2 for 1-10 parallel streams as shown in Figure 3(a)-(b). They show a decreasing trend as rtt is increased and at a fixed rtt they show an increasing trend as the number of parallel streams is increased. Since XDD uses TCP for wide-area transport, these throughputs represent upper limits on file transfer rates. Memory transfer rates are measured using XDD for single and 8 IO threads using different request or block sizes as shown in Figure 3 (c) and (d), respectively. The results show that for a single stream, XDD transfer rates are lower than TCP throughput and varied based on the request size, and interestingly the lowest request size provided peak transfer rates. When 8 streams are employed, the memory transfer rate of XDD closely matches iperf throughput, which indicates that the overheads introduced by XDD for transfers are rather limited. When file systems are engaged at the end hosts, the lower xfs file transfer rates in Figure 1 are due to the file IO rather than the effects of transfer overheads introduced on host systems by XDD.

IV. Host File IO throughput

XDDprof tool is used to measure file IO throughput rates on host systems by reading and writing files for different parameters such as the number of streams and request size; other parameters such as direct and random IO policies are not utilized in our measurements. xfs and Lustre file systems provided peak file IO throughput much above 10Gbps, which are above the peak TCP throughput over the connections described in the previous section. Thus, the capacity of file IO on these systems is not a limiting factor for the end-to-end file transfer rates.

V. File Transfer Rates

A summary of average file and memory transfer rates achieved by XDD is shown in Figure 4 for 8 IO threads which also correspond to 8 parallel TCP streams. Both memory and xfs transfer rates are comparable to the corresponding TCP throughput as shown in Figure 4(a) and (c), respectively. Furthermore, the request sizes did not have a significant effect on the transfer rate profiles. Lustre transfer rates are lower than xfs memory transfer rates, and they also showed an increasing trend with rtt along with significant variations when the request size is varied. The transfer rates for nfs are in between and showed a decreasing trend with respect to rtt but varied with the request size to a lesser degree compared to Lustre.



(a) memory transfer

(b) Lutre file transfer

(c) xfs read file transfer

(d) nfs file transfer

Fig. 4. Measurements of XDD transfer rates for memory and lustre, xfs and nfs file systems.

A. *Lustre Transfer Rates*

A detailed analysis of Lustre transfer rates indicates trends that are quite different from those observed in other cases, namely TCP throughput, XDD memory transfer rates and xfs file transfer rates.

(a) **Lower Transfer Rate:** Highest transfer rate around 3 Gbps is achieved using a single stream and the smallest request size of 8192 bytes. Furthermore, the transfer rate decreased with rtt when a single stream is used as shown in Figures 5 (c), (e) and (g). However, the rate profile is lowered as the request size is increased, and the reduction was more than 50% for rtt higher than 11ms for 144k request size as shown in Figure 5(g).

(b) **Increasing Transfer Rate Profiles:** When 8 streams are utilized, Lustre transfer rates are much lower, typically below 2Gbps but exhibit an increasing trend with respect to rtt. Furthermore, this increasing trend is more pronounced as the request size is increased, and also the transfer rates varied rather drastically when repeated. As a result it is harder to accurately predict the transfer rates for these parameters.

B. *xfs Transfer Rates*

We collected XDD transfer rate measurements separately for read and write operations as shown in Figures 6 and 7, respectively. A detailed analysis of the measurements indicates that both read and write profiles are qualitatively consistent with typical trends observed in TCP throughput and XDD memory transfer rates.

(a) **Higher Transfer Rates:** Transfer rate around 7 and 9 Gbps are achieved using single and 8 streams, respectively, for lower rtt. The request size did not have much effect on read rates but the peak write rate is achieved with a small request size. These peak rates are within 10% of peak TCP throughput for the corresponding rtt, and file IO did not seem to severely constrain the peak transfer rates.

(b) **Decreasing Transfer Rate Profiles:** In case of single and 8 streams, xfs read and write transfer rates exhibit decreasing trends with respect to rtt; however, multiple streams additionally exhibited wider concave profiles compared to mostly convex profiles in single flow cases. Also, XDD read rates are slightly higher than those of XDD write, and they also exhibited somewhat smaller variations.

Thus, the overall XDD parameters that achieve peak xfs transfer rate are different from those for Lustre in requiring more streams but similar in requiring a smaller request size.

VI. Conclusions

We collected network and file IO throughputs for Lustre and xfs file systems over dedicated 10Gbps connection with 0-366ms rtt to gain a qualitative understanding of the underlying parameters and their optimizations. The measurements indicate that file IO and TCP transport parameters must be jointly selected to achieve peak file transfer rates, and in some cases these parameters could be significantly different from those that achieve peak throughput for individual file IO and TCP transfers. Over our testbed, peak rates for Lustre transfers are achieved using a small request size and single IO and TCP stream, which is in sharp contrast with xfs file transfers that achieved peak rates with eight IO and TCP streams. Furthermore, xfs and memory transfer rates approach the connection capacity for smaller rtt, and they generally decrease for larger rtts. But Lustre file transfer rates are much lower than the connection capacity and showed a surprising increasing trend with increasing rtt along with high statistical variations. While these observations are specific to our testbed, they do indicate the need for careful joint optimization of file IO and TCP transport parameters.

Future work includes testing other parameters and configurations for network transport including TCP congestion control versions, such as Scalable TCP and Hamilton TCP, and UDP based protocols such as UDT. For file IO, future testing could include various read/write policies including direct IO. It would be of future interest to explore efficient in-situ automated methods to jointly search for file IO and network transport parameters that achieve peak transfer rates without sweeping through the entire parameter space.

Acknowledgments

This work is supported in part by the United States Department of Defense and used resources of the Computational Research and Development Programs, and is also supported in part by Net2013 and RAMSES projects, Office of Advanced Computing Research, Department of Energy at Oak Ridge National Laboratory managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725.

References

[1] On-demand secure circuits and advance reservation system. http://www.es.net/oscars.

[2] Science DMZ: Data Transfer Nodes, https://fasterdata.es.net/science-dmz/DTN.

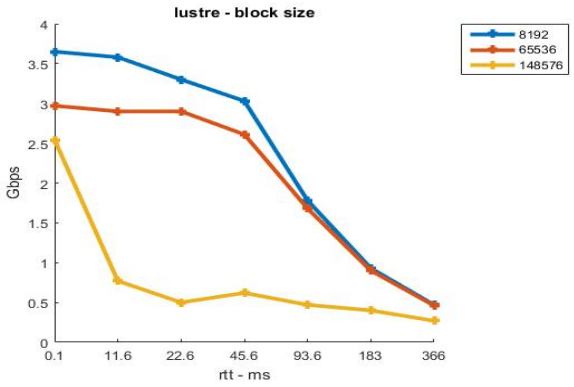
[3] Lustre Basics, https://www.olcf.ornl.gov/kb_articles/lustre-basics.

[4] 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*, 2015.

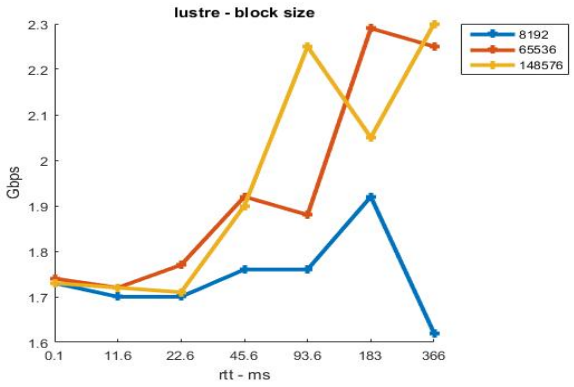
[5] B. W. Settlemyer, N. S. V. Rao, S. W. Poole, S. W. Hodson, S. E. Hicks, and P. M. Newman. Experimental analysis of 10gbps transfers over physical and emulated dedicated connections. In *International Conference on Computing, Networking and Communications*, 2012.

[6] Linux tuning, https://fasterdata.es.net/host-tuning/linux.

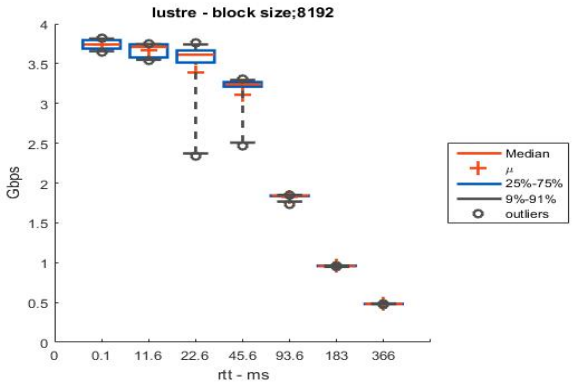
[7] XDD - The eXtreme dd toolset, https://github.com/bws/xdd.



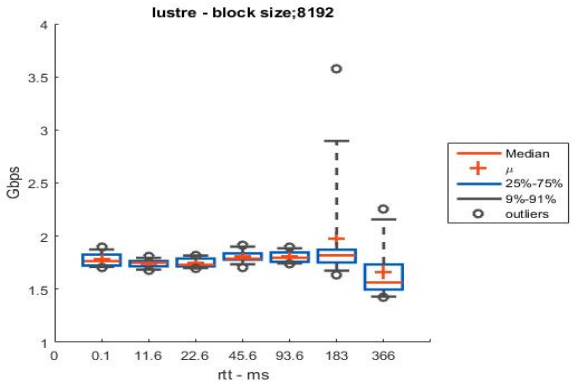
(a) single stream- 8k request size



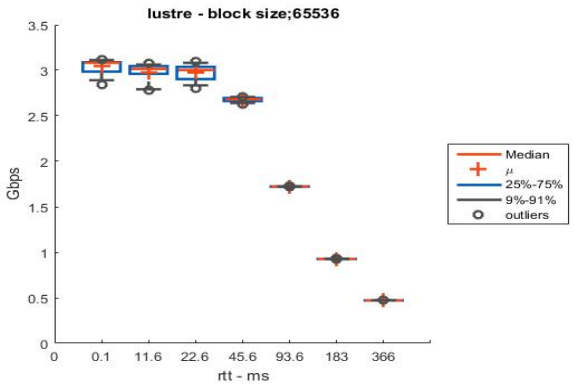
(b) 8 streams - 8k request size



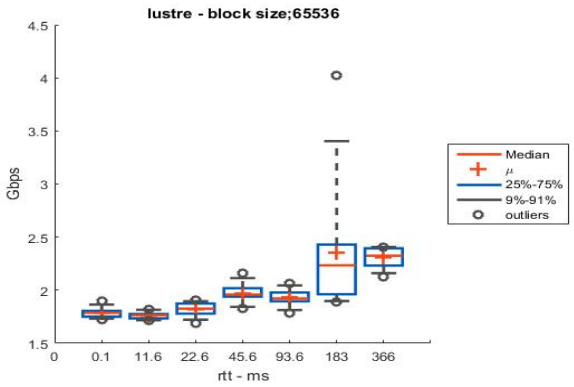
(c) single stream- 8k request size



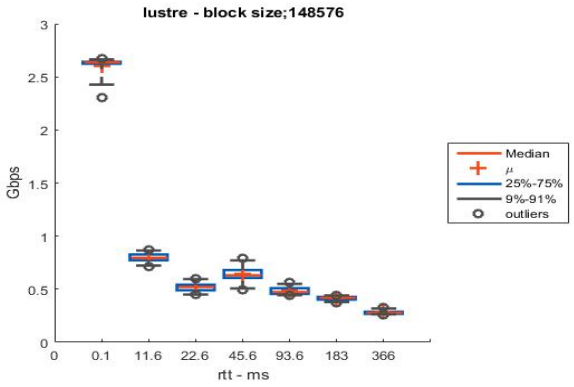
(d) 8 streams - 8k request size



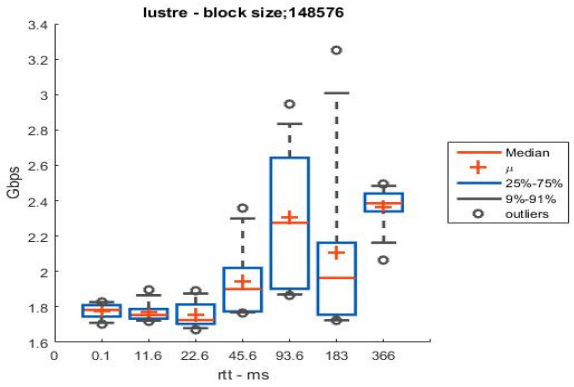
(e) single stream - 65k request size



(f) 8 streams - 65k request size

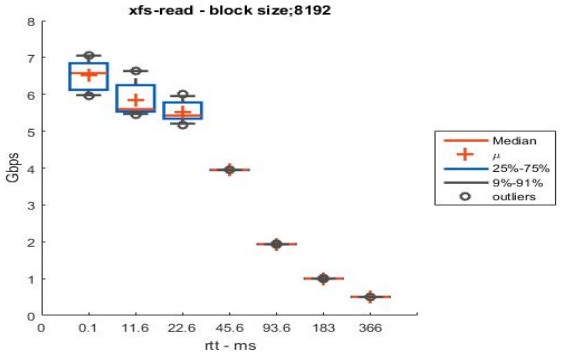


(g) single stream - 144k request size

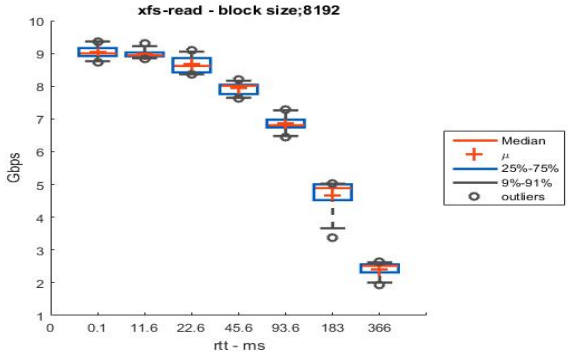


(h) 8 streams - 144k request size

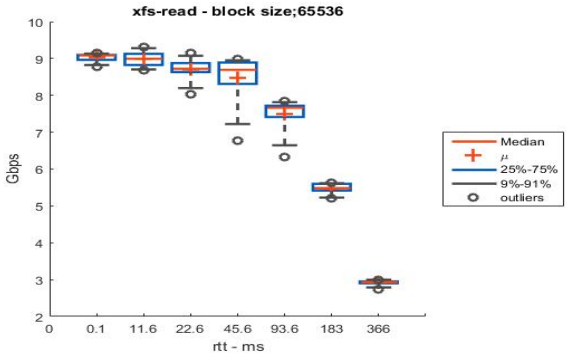
Fig. 5. Measurements of Lustre transfer rates for different rtt, request sizes for single and 8 streams.



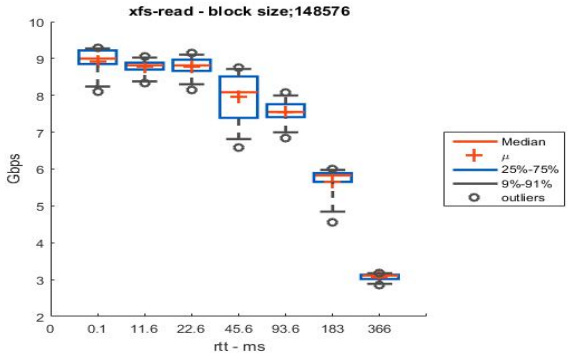
(a) single stream- 8k request size



(b) 8 streams - 8k request size

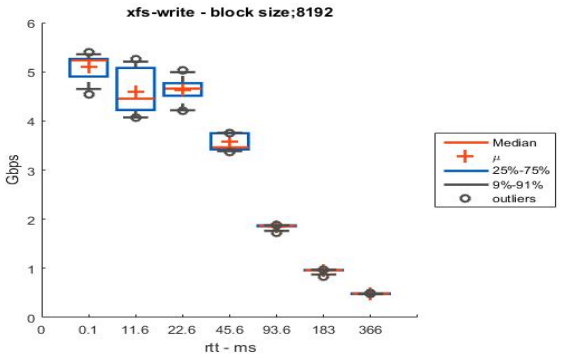


(c) 8 streams - 65k request size

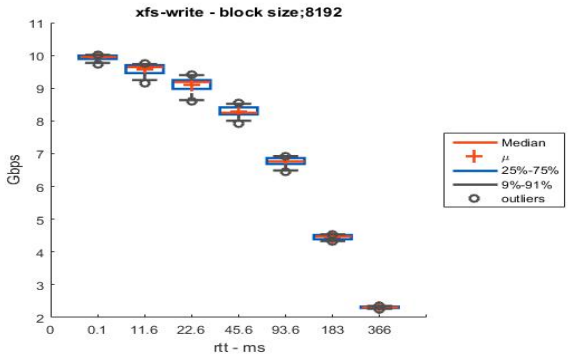


(d) 8 streams - 144k request size

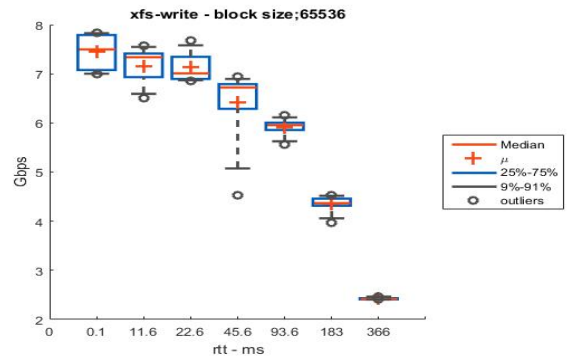
Fig. 6. Measurement of xfs file read transfer rates for different rtt, request sizes for single and 8 streams.



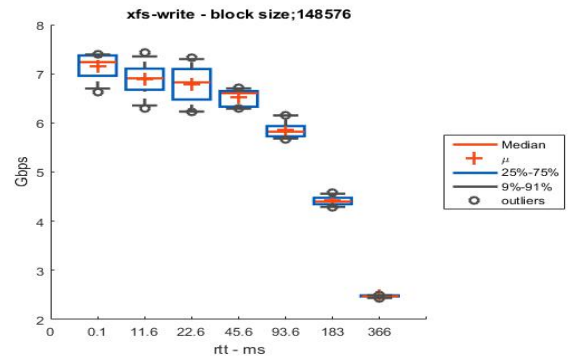
(a) single stream- 8k request size



(b) 8 streams - 8k request size



(c) 8 streams - 65k request size



(d) 8 streams - 144k request size

Fig. 7. Measurement of xfs file write transfer rates for different rtt, request sizes for single and 8 streams.