# Identification of Network Data Transfer Bottlenecks in HPC Systems

Karen Tu
University of California, Berkeley
karentu@berkeley.edu

## ABSTRACT

Improving network data transfer performance is a major factor for improving high performance computing systems. Most studies analyze data transfer and file system IO performance separately, but understanding the relationship between the two is essential for optimizing scheduling and resource management. Intuitively, if data is being transferred to a busy file system the transfer rate would be slower than a file system at regular activity levels. This study analyzes patterns between file system activity and network throughput for several use cases of file writing and data transfers using a parallel file system. The parameters changed among the use cases were file striping for the file system, and buffer size and parallelism for data transfer. The main bottleneck for network data transfer rate was the number of OSTs the data was striped across. For a large number of OSTs (16 or greater), writing to the file system was the bottleneck.

## 1 INTRODUCTION

One of the major factors for improving the efficiency and scalability of high performance computing (HPC) systems is improving performance of file storage and data transfer across the network. Large volumes of data are transferred between HPC facilities, and identifying bottlenecks in storage I/O performance and network throughput performance is essential for optimizing the process. There are multiple studies about on improving performance of network data transfers in HPC systems alone, such as reducing throughput variance [1], but this research is focused on improving the combined network and storage system performance. It is likely that high file IO rates are correlated with lower network transfer rates because if a file system is already busy, network transfers to that file system should have decreased performance. If this is a problem for an HPC system, it might be better to schedule these two activities at different times to optimize resource usage.

This work is experimental and the data collected are from the CORI supercomputer at NERSC. Similar to Zhengchun Liu et. al's work on wide area network data movement [2] and Jinoh Kim et al.'s work on HPC file IO bottlenecks [3], by incorporating different levels of system logs, this research involves both file system and network data logs for a more comprehensive view of data transfer characteristics.

## 2 MATERIALS AND METHODS

The Lustre file system connected to the CORI supercomputer at NERSC was used to perform file writes. With 248 OSTs (object storage targets), the file system has an aggregate peak IO performance of 744 GB/second. Log files containing read and write rates for each OST were collected using LMT (Lustre monitoring tool) [4].

To test data transfer throughput rates, two of the ten DTNs (data transfer nodes) at NERSC were used. The Globus file transfer tool was used to transfer data through the Lustre file system and DTNs as well as for calculating throughput.

Several use cases with varying Lustre striping parameters (stripe size and number of OSTs to put file on) and data transfer parameters (buffer size and parallelism) were used to determine if high file system activity negatively impacts data transfer rates. The start and end times of each data transfer were used to identify exactly which log files were associated with the experiment. Then read/write amounts were aggregated across all OSTs and studied their correlation with the network throughput.

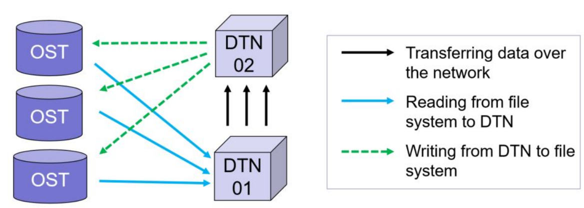



**Figure 1: Data transferred between filesystem OSTs (object storage targets) using the DTNs (data transfer nodes)**

For each use case, three sets of data were collected:

1. Read from file system + transfer over network
2. Transfer over network + write to file system
3. Read from file system + transfer over network + write to filesystem

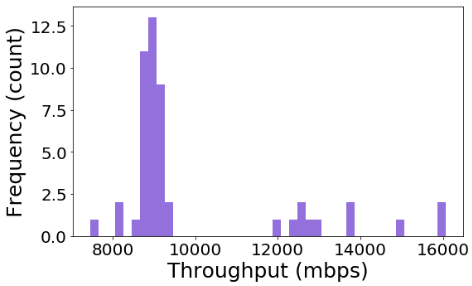



**Figure 2: Throughput distribution for network data transfer**

Benchmark values for throughput for data transfer only through the network without interacting with the file system were taken by measuring throughput of transferring data between two DTNs. Figure 2 shows the distribution with an average of 9927.44 mbps, and standard deviation of 2102.52 mbps.

## 3 RESULTS

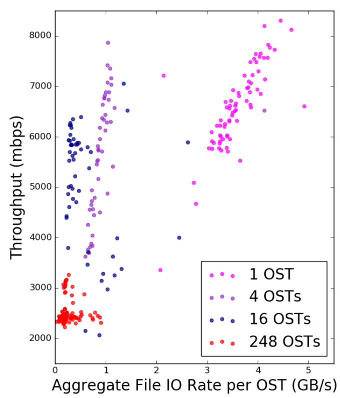
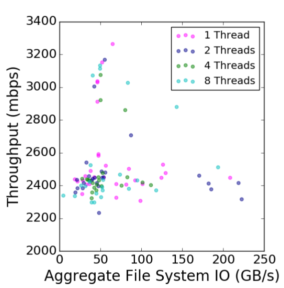



**Figure 3: Total file IO activity per OST vs. throughput**

According to figure 3, the number of OSTs a file is striped across has the biggest impact on throughput. The default of one OST yields the best performance while 248 OSTs yields the worst.



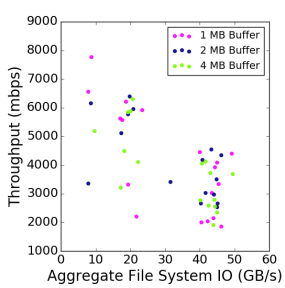




**Figure 4: File system activity vs. throughput for writing to a file striped across 248 OSTs with different parallelism settings**

**Figure 5: File system activity vs. throughput for reading a file striped across 16 OSTs with different buffer size settings**

Varying file stripe size in the file system and the buffer size and parallelism during data transfers had no effect on throughput rates (figures 4, 5). Different combinations of those parameters and holding certain values constant yielded equally random scatter plots; there were no patterns.

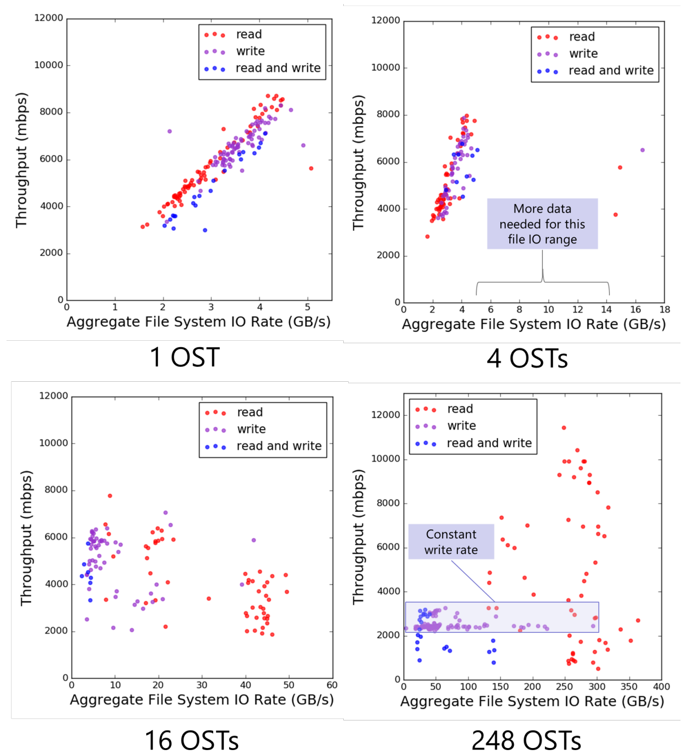



**Figure 6: Total file IO activity (corresponding to the OSTs that the file is striped across) vs. throughput of different types of file transfers**

Intuitively, if a file system is busy with reading and writing, then the throughput of data transfers to that system should be lower compared to if the file system were at regular activity levels. However, from the data this is not generally the case. For files stored on one or four OSTs, the throughput goes up as the file system gets busier. For 16 or more OSTs, as the file system gets busy there is a downward trend in throughput (figure 6). For files stored in 248 OSTs the write rate is constant and lower than the average read rate (figure 6), which indicates that writing to the file system is a data transfer bottleneck.

## 4 CONCLUSIONS

A higher number of OSTs is a bottleneck in network data transfer rates. Counterintuitively, for fewer OSTs, there is an association between high file system activity and high data transfer throughput. This may be related to the performance of the OSTs; when there is low file activity it may be the OSTs that are the bottlenecks and not just that the OSTs are not being used to transfer any data.

The high variation in throughput values overall suggests there are other parameters affecting throughput not considered in this study. Peak data transfer rates do not seem related to file system stripe size, data transfer buffer size, and data transfer parallelism which are the usual parameters tuned to improve network throughput.

## 5 FUTURE WORK

In the future, there will be research done with the storage team to conduct case studies to control for file system activity and determine if OST performance could be a bottleneck in network throughput. A similar analysis will be done using different file sizes and structures.

## ACKNOWLEDGMENTS

I would like to thank my mentors Alexander Sim and John Wu for their guidance and feedback. Special thanks goes to Prof. Jinoh Kim and Mr. Jinhwan Kim at Texas A&M, Commerce for their guidance on methods. This work was supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internship (SULI) program. This work was supported by the U.S. Department of Energy, under Contract No. DE-AC02-05CH11231 and used resources of the National Energy Research Scientific Computing Center.

## REFERENCES

[1] Z. Liu, M. Veeraraghavan, J. Zhou, J. Hick, and Y. Li, "On Causes of GridFTP Transfer Throughput Variance", in "Proceedings of the Third International Workshop on Network-Aware Data management (NDM '13)", 2013, 5-10, doi: 10.1145/2534695.2534701

[2] Z. Liu, R. Kettimuthu, I. Foster, and Y. Liu, "A comprehensive study of wide area data movement at a scientific computing facility", 2018.

[3] J. Kim, J. Choi, A. Sim, "Spatio-temporal Analysis of HPC I/O and Connection Data", International Workshop on Scalable Network Traffic Analytics (SNTA 2018), 2018, in conjunction with the 38th IEEE International Conference on Distributed Computing Systems (ICDCS 2018), 2018.

[4] Wartens, H. C. M. & Garlick, J. LMT - The Lustre Monitoring Tool. https://github.com/chaos/lmt/wiki/. Developed at Lawrence Livermore National Lab. (2010)