Identification of Network Data Transfer Bottlenecks in HPC Systems

Karen Tu[1], Alexander Sim[2], John Wu[2]
[1]University of California Berkeley, [2]Lawrence Berkeley National Laboratory

U.S. DEPARTMENT OF ENERGY
Office of Science

INTRODUCTION

- High performance computing systems are used to store and transfer large volumes of data - identifying bottlenecks in data transfer is essential for optimizing performance
- Data transfer and file system IO rates are often analyzed separately; this study looks at the relationship between them

MATERIALS & METHODS

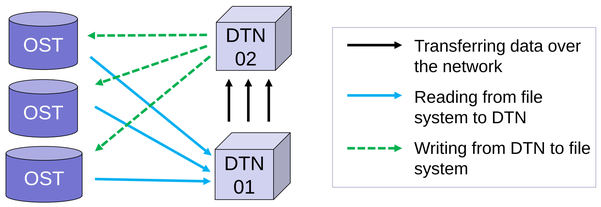

Figure 1. Data is transferred to/from the OSTs (object storage targets) of the file system using the DTNs (data transfer nodes)

- Throughput Data Collection

1. read from file system + network transfer
2. Network transfer + write to file system
3. Read from file system + network transfer + write to file system

*Throughput: Rate of data transfer over network

- Lustre file system: 248 OSTs (object storage targets), aggregate IO peak performance : 744 GB/s

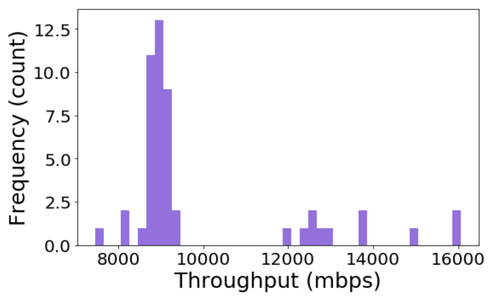

Figure 2. Distribution of throughput values for network data transfer

- Transfer rate between data transfer nodes
  - Average: 9927.44 mbps
  - Standard deviation: 2102.52 mbps

File System Activity Tools

- Lustre monitoring tool: log files of file system IO rates

Network Data Transfer Tool

- Globus: gridftp protocol to transfer data between data transfer nodes and file system

RESEARCH QUESTION

If a file system is busy with reading and writing, will this negatively impact the data transfer rate of data being transferred into that file system?

RESULTS

The number of OSTs a file is striped across is a strong indicator of its throughput

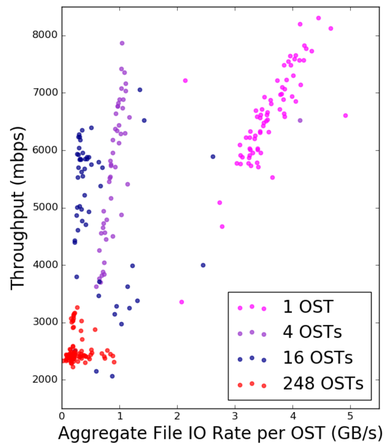

Figure 3. Total file IO activity per OST vs. throughput

- 1 OST: high file IO correlated with high throughput
- 16 OSTs: high file IO correlated with lower throughput
- 248 OSTs: writing is the bottleneck

File system activity vs. throughput for various levels of file striping and interaction with the file system (only reading, only writing, or both reading and writing)

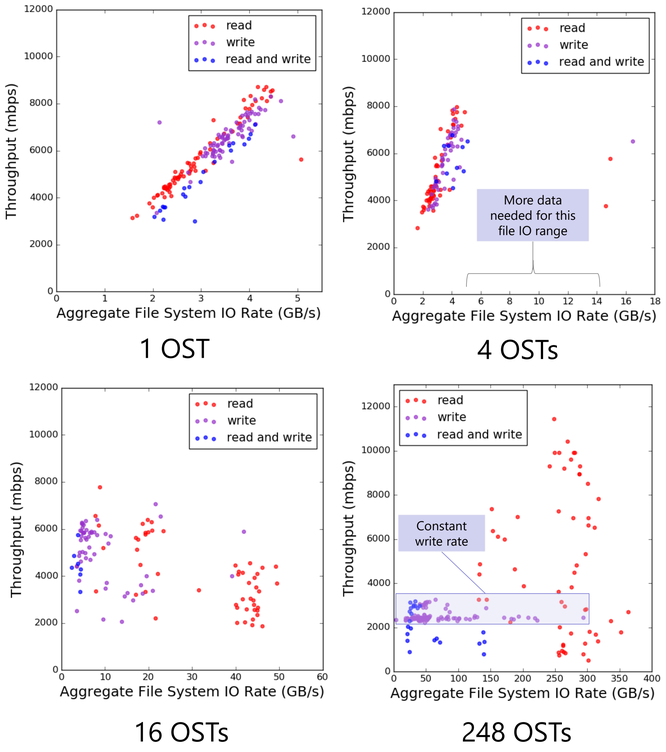

Figure 4. Total file IO activity (corresponding to the OSTs that the file is striped across) vs. throughput of different types of file transfers

No pattern relating throughput to file stripe size in file system

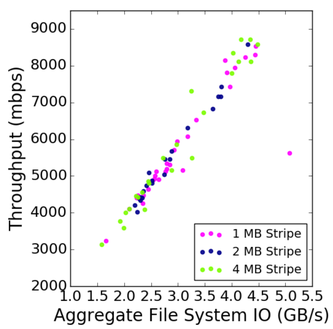

Figure 5. File system activity vs. throughput for reading a file striped across 1 OST with different stripe settings

No pattern relating throughput to buffer size or number of parallel streams during network data transfer

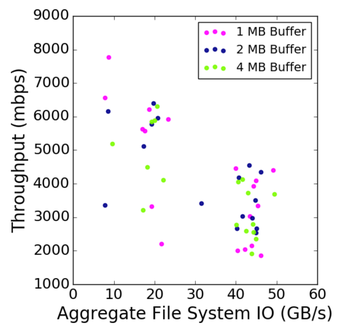

Figure 6. File system activity vs. throughput for reading a file striped across 16 OSTs with different buffer size settings

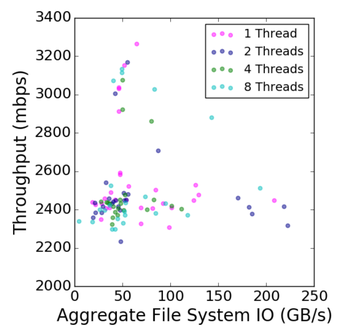

Figure 7. File system activity vs. throughput for writing to a file striped across 248 OSTs with different parallelism settings

DISCUSSION

- If a file system is busy with reading/writing, throughput of data transfers to that system should be lowered – this is not the case
- Throughput is related to the number of OSTs a file is striped across
  - Default of 1 OST yields best performance, 248 OSTs yields worst performance
- Bottlenecks for network data transfers occur in different places depending on number of OSTs
  - 248 OSTs: writing
  - 1, 4, or 16 OSTs: reading
- Tuning various file system and data transfer parameters had no effect on throughput rates
  - Combinations of those parameters and holding certain values constant yielded similar looking scatterplots; no patterns

CONCLUSION

- High variation in throughput values suggest there are other parameters affecting throughput not considered in this study
- A higher number of OSTs is a bottleneck in network data transfer rates
- While file system stripe size and data transfer buffer size and parallelism were not bottlenecks, it is odd that there was no way to tune them to increase transfer performance

FUTURE WORK

- Work with file system team to run controlled case studies
  - Control for file system activity
- Collect data using different file sizes/structures
- Acquire more information about DTNs

ACKNOWLEDGMENTS

I would like to thank my mentors Alexander Sim and John Wu for their guidance and feedback. 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.