

# Spatio-temporal Analysis of HPC I/O and Connection Data

Jinoh Kim\* Jinhwan Choi\* Alex Sim†

Texas A&M University, Commerce, TX 75428, USA \*

Lawrence Berkeley National Laboratory, Berkeley, CA 94720 †

Email: jinoh.kim@tamuc.edu, jchoi8@leomail.tamuc.edu, asim@lbl.gov

**Abstract**—The HPC system consists of a set of layers of software and hardware for I/O and networking. System logs are helpful resources to understand what is going on in the system. A challenge is that it is non-trivial to analyze the logs maintained in various levels of the stack. Independent analysis might lead to an incomplete conclusion due to the limited coverage of each log. This work takes a comprehensive approach to analysis that incorporates the logs in the multiple layers and components, in order to facilitate the detection of anomalous activities. This research aims to identify and predict potential performance bottlenecks in the HPC system, by capturing the temporal variation patterns from heterogeneous, high-dimensional, and non-linear log data. In this paper, we share our preliminary efforts for spatio-temporal analysis of HPC I/O and connection data, with our initial observations from the analysis of one-week HPC log data sets collected from one of NERSC systems.

## I. INTRODUCTION

High-performance computing (HPC) systems have achieved significant advancements in capacity and scale. At the same time, the complexity of the system has also considerably increased, which makes it hard to identify and predict performance bottlenecks and anomalous events impacting performance and reliability. In particular, data access and sharing has become more important for many HPC applications in the big data era. For example, a single cosmology application generates 20 PB of data, which is exchanged among dispersed computing facilities [1]. Identifying I/O and network bottlenecks should thus be one of the first-class requirements to improve efficiency and scalability in HPC systems.

The HPC system consists of a set of layers of software and hardware for I/O and networking. System logs are helpful resources to understand what is going on in the system. For example, Luster Monitoring Tool (LMT) records I/O activities to a log, while the network transfer activities are logged using various tools such as NetFlow. A challenge is that it is non-trivial to analyze the logs maintained in various levels of the stack. Independent analysis might lead to an incomplete conclusion due to the limited coverage of each log. For example, the CPU load of meta-data server (MDS) may affect the I/O performance of a specific HPC application [2]. This work takes a comprehensive approach to analysis that incorporates the logs in the multiple layers and components, in order to facilitate the detection of anomalous activities.

This research aims to identify and predict potential performance bottlenecks in the HPC system, by capturing the temporal variation patterns from heterogeneous, high-dimensional,

and non-linear log data. Recently, data analysis has been considerably improved with the disruptive advances of machine learning technologies. However, we still see several challenges. One challenge is that the increasing volume of data would be a critical barrier to analyze them in a comprehensive manner. In addition, many learning-based techniques assume the presence of the associated labels to analyze; however constructing labels is highly laborious with human expert support, and hence not available in practice. To attain the research goal, there is no doubt that employing machine learning technologies will be essential, but it may not be easy to simply apply learning methods to analyze the log data in the system due to such challenges.

In this paper, we share our preliminary efforts for spatio-temporal analysis of HPC I/O and connection data. Rather than simply relying on machine learning, we manually analyzed the log data to find out potential behavioral correlations that could cause performance bottlenecks. We present our initial observations from the analysis of one-week (January 1–6, 2018) HPC log data sets collected from a NERSC Cori HPC system<sup>1</sup>. The data set in this study includes the metadata server CPU load, file system logs sampled once every 5 seconds, and TCP connection records.

This paper is organized as follows. The following section provides a summary of the related studies on performance bottleneck problems in HPC environments. Section III describes our method to analyze the HPC log data with the overview of the HPC system model. In Section IV, we share our observations from the analysis work. Finally, we conclude our presentation in Section V with a summary.

## II. RELATED WORK

There has been a substantial body of work to identify performance problems in a cluster computing environment. The authors in [3] used fingerprints generated from micro-benchmarks. to identify performance bottlenecks through machine learning techniques. The work in [4] relies on predefined rules to identify performance bottlenecks of HPC applications. In [5], the authors designed a tool that correlates hardware events with source code to identify performance bottlenecks of parallel applications. Another work [6] presented a variable selection technique based on performance prediction to enable scalable log analysis.

<sup>1</sup><http://www.nersc.gov/users/computational-systems/cori/>

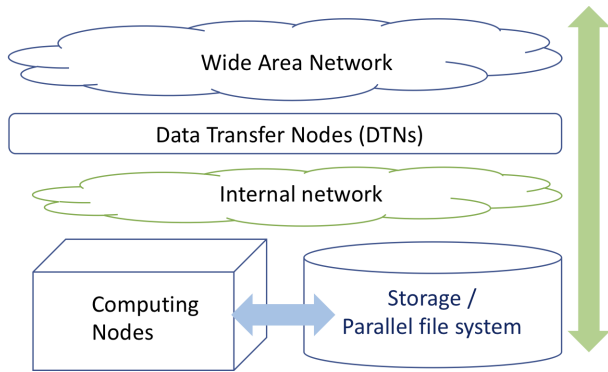


Fig. 1. Data flows view to the HPC storage

PATHA [7] has been developed to classify performance characteristics and also to identify performance bottlenecks for scientific workflow applications in the HPC environment. To figure out time-consuming operations in the application, this tool performs the execution time analysis. In addition, PATHA also provides a function for data dependency performance analysis to figure out data dependencies that may cause performance bottlenecks. The focus of this past work is on providing users with an interactive tool for promoting the users understanding about the performance of their applications.

Another work in [2] analyzed performance bottlenecks in the HPC system for a specific application of Adaptive Mesh Refinement (AMR). From the analysis of the parallel I/O subsystem logs, this past work found some bottleneck points that reduce I/O bandwidth significantly by up to 40 times. All of these past studies are closely related to our research, but our interest is more on the system-wide analysis (rather than the focus of the HPC application performance). In addition, our work incorporates the I/O and network data to expect a greater degree of comprehensiveness in analysis.

### III. METHODS

This study focuses on the wide area network data transfer throughput performance and the storage I/O performance based on the parallel file system. Scientific data centers typically consists of networking nodes for the wide area data transfers, with parallel file systems in the backend. When the storage systems are shared with other HPC jobs for data I/O, the data transfer throughput performance through the networking nodes get affected by the other HPC I/O activities. Figure 1 shows a high-level view of data flows to the HPC storage with wide area network data transfer traffic and local data traffic with computing nodes [8]. Data Transfer Nodes (DTNs) as a part of the Science DMZ are dedicated systems for wide area data transfers with special configuration and proper transfer tools for high throughput performance. All DTNs at NERSC have four 10-gigabit Ethernet links for wide area network and two FDR IB connections to the each underlying file system.

Figure 2 shows a simplified view of Lustre file system, one of the parallel file systems at NERSC with about 30PB of

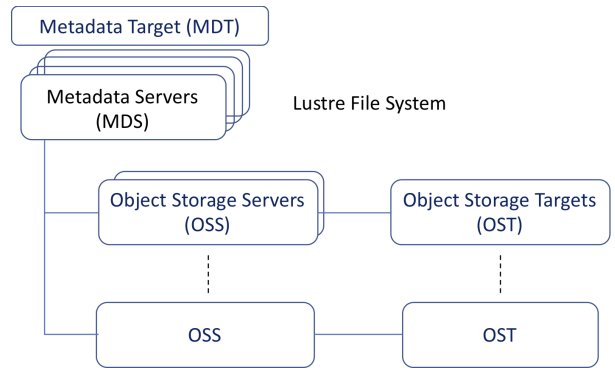


Fig. 2. Simplified view of Lustre file system

storage with 248 Object Storage Targets (OSTs). The Lustre file system is composed of a set of IO servers and disks called Object Storage Servers (OSSs) and Object Storage Targets (OSTs), respectively. A file is usually striped for read and write operations accessing multiple OSTs concurrently to increase I/O performance.

This work studies on integrating performance logs from two key levels of parallel I/O subsystems from Cori at NERSC: application-level POSIX and Message Passing Interface (MPI) I/O traces, coming from the Darshan logs, and activity logs from LMT, a monitoring tool for the OSTs and MDSs of the Lustre parallel file system. The network traffic data set contains the `tstat` logs, collected to analyze how various network tuning settings impact TCP behavior and network throughput. `tstat` rebuilds each TCP connection by looking at the TCP header in the forward and reverse direction. The details about the `tstat` tool can be found in [9].

### IV. EXPERIMENTAL RESULTS

In this section, we report our initial observations learned from the analysis of the traces collected between January 1–6, 2018 from the Cori system. Figure 3 shows the OST I/O activities the network throughput over the six days. To see if there is any potential performance bottleneck, we also monitored the MDS CPU load with the intuition that storage I/O and network transmission might be slowed down if the meta-data server is busy to service the requests (or due to something else internally).

We observed that the MDS CPU load is less than 60% over 99% of the collection period. In this analysis, we assumed that MDS is busy if the CPU load is greater than 60%. In Figure 3, the solid dots indicate the time points at which the MDS CPU load is equal to 60% or above. As can be seen from the figure, storage read/write rates are quite small in case that MDS is busy. It is almost the same for network throughput, but we also observed a couple of cases that the throughput is higher than the average even in the case of MDS busy.

To see if there is any impact to data access rate and network throughput once MDS is busy, we closely look at the time intervals when the MDS CPU usage is greater than 60%. Figure 4 demonstrates the monitored result in the interval from

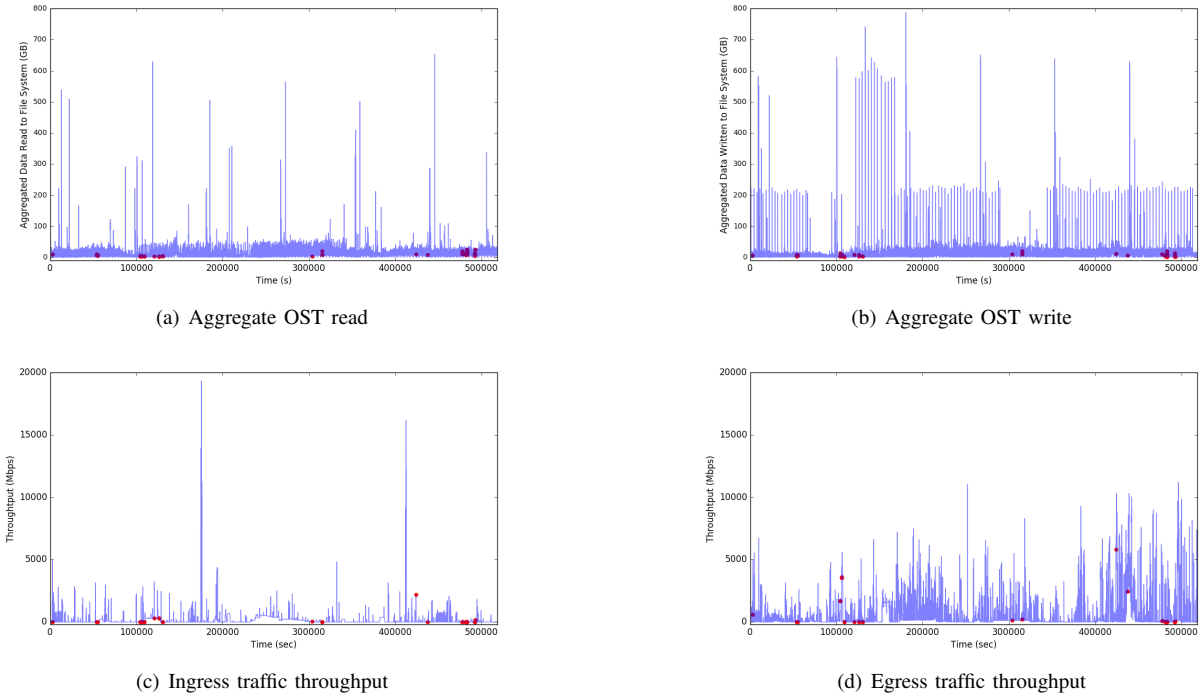


Fig. 3. Aggregated OST read/write and tstat ingress/egress traces over six days. The solid dots indicate the time points at which the MDS CPU load is equal to 60% or above.

$t=55100$  to  $t=55200$ . Figure 4(a) shows the MDS CPU load and aggregated read/write rates. From the figure, we can see that MDS is busy in  $t=20-35$ . Within that busy interval, it is observed that the storage read rate decreases quickly but gets back once the MDS load goes down at  $t=40$ . The write pattern also shows the access rate goes higher at  $t=35$  which is the starting point that the MDS load goes down.

To see further details, Figure 4(b) and 4(c) show the individual OSTs read and write activities, respectively, in terms of  $\Delta = \text{current window data rate} - \text{past window data rate}$  (window size = 5 seconds). Thus, the positive indicates the increase of data rate compared to the past window, and vice versa. From the figures, we can see that the majority of OSTs changed to a downturn at  $t=5$  and  $t=35$ , respectively, at which the CPU usage is relatively high.

Figure 4(d) and 4(e) demonstrate the individual OSTs data access contributions. We selected a subset of OSTs to demonstrate since there are too many OSTs (248 OSTs in total). The selected OSTs have at least 150 MB in any time window over the time interval. In the contribution graph, one bar indicates an aggregated data rate of the selected OSTs in a 5-second window; that is, the first bar shows the data rate observed within the interval of  $t=0-4$ . For read, we can see that the OST data rate is significantly reduced from  $t=25-29$  to the next time interval of  $t=30-34$ . The data rate decreased once again in  $t=40-44$  (right after the second CPU peak at  $t=35$ ). For write, it shows a significant downturn in  $t=25-29$ , but no critical change is observed at the second CPU peak.

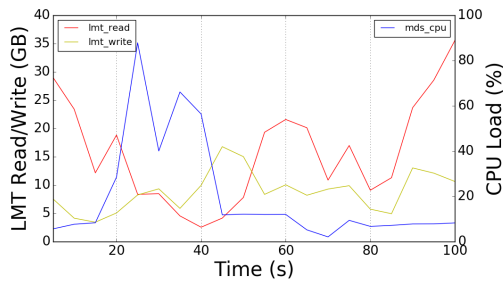
Unlike the OST read and write, we observed little impact of the MDS load to the ingress and egress network throughput

in any time interval.

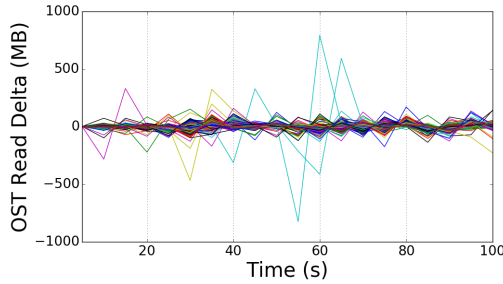
Figure 5 shows another example at  $t=126220-126320$ , during which there are two CPU usage peaks at  $t=45$  and  $t=55$ . From the figure, we can see that write throughput changes inversely against the CPU usage in the peak time interval ( $t=40-60$ ); that is, write throughput significantly decreased at the first peak interval, goes up at  $t=50$ , and then goes down again at the second peak. For read throughput, there is little change in the first peak, but it goes up at  $t=50$  when the CPU usage goes down. Then read throughput is also reduced after the second peak. Thus, this interval also shows a degree of correlation among the MDS load and I/O rates.

## V. CONCLUSION

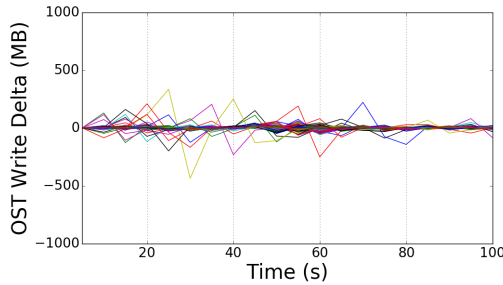
Identifying and predicting performance bottlenecks in the HPC system are the crucial concern. For this, capturing the temporal variation patterns is an essential task from heterogeneous, high-dimensional, and non-linear log data. As an initial study, we analyzed HPC I/O and connection data to understand the potential correlation among MDS CPU usage, I/O rates, and network traffic throughput. We observed that the MDS load may affect the I/O rates from the temporal patterns, while network throughput does not show any clear correlation. We also observed some OSTs showed a degree of correlations with the MDS load patterns in an inversely proportional fashion. This work is an on-going work and still in the initial stage. We plan to extensively analyze a long term data spanning months to collect the indication of correlations among different logs and data sources.



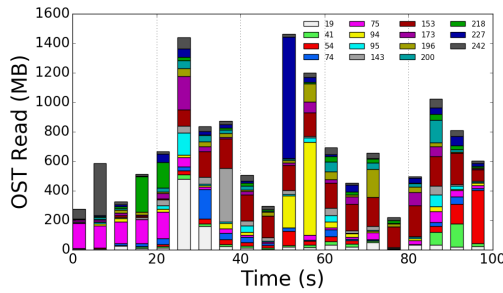
(a) Metadata server CPU usage and aggregated I/O rates



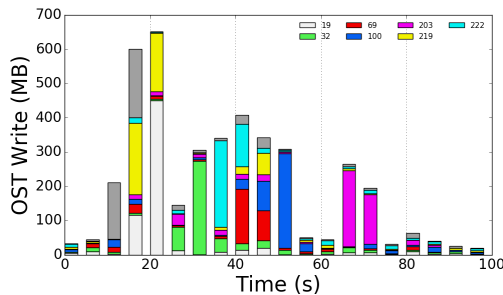
(b) OST read delta



(c) OST write delta



(d) OST read contribution



(e) OST write contribution

Fig. 4. A closer look within the interval of  $t=55100-55200$

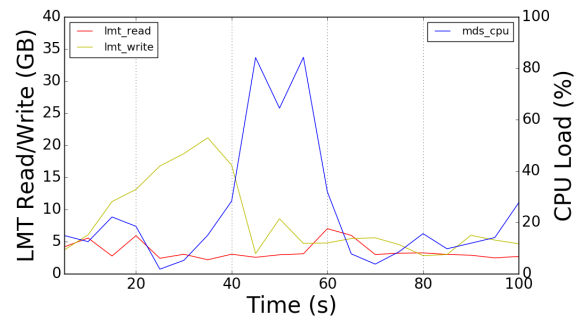


Fig. 5. Metadata server CPU usage and aggregated I/O rates at  $t=126220-126320$

## ACKNOWLEDGMENT

This work was supported by the Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research used resources of the National Energy Research Scientific Computing Center.

## REFERENCES

- [1] Rajkumar Kettimuthu, Zhengchun Liua, David Wheeler, Ian Foster, Katrin Heitmann, and Franck Cappello. Transferring a petabyte in a day. In *IEEE/ACM SC17 Workshop on Innovating the Network for Data Intensive Science (INDIS 2017)*, 2017.
- [2] Peter Harrington, Diagnosing Parallel I/O Bottlenecks in HPC Applications, International Conference for High Performance Computing, Networking, Storage and Analysis (SC17), ACM Student Research Competition (SRC), 2017.
- [3] Wucherl Yoo, Kevin Larson, Lee Baugh, Sangkyum Kim, and Roy H. Campbell. Adp: Automated diagnosis of performance pathologies using hardware events. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '12, pages 283–294, 2012.
- [4] Martin Burtscher, Byoung-Do Kim, Jeff Diamond, John McCalpin, Lars Koesterke, and James Browne. Perfexpert: An easy-to-use performance diagnosis tool for hpc applications. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '10, pages 1–11, 2010.
- [5] L. Adhianto, S. Banerjee, M. Fagan, M. Krentel, G. Marin, J. Mellor-Crummey, and N. R. Tallent. Hpctoolkit: Tools for performance analysis of optimized parallel programs <http://hpctoolkit.org>. *Concurr. Comput. : Pract. Exper.*, 22(6):685–701, April 2010.
- [6] Jonathan Wang, Wucherl Yoo, Alex Sim, Peter Nugent, and Kesheng Wu. Parallel variable selection for effective performance prediction. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, CCGrid '17, pages 208–217, 2017.
- [7] Wucherl Yoo, Michelle Koo, Yi Cao, Alex Sim, Peter Nugent, and Kesheng Wu. PATHA: performance analysis tool for HPC applications. In *34th IEEE International Performance Computing and Communications Conference, IPCCC 2015, Nanjing, China, December 14-16, 2015*, pages 1–8, 2015.
- [8] Eli Dart, Lauren Rotman, Brian Tierney, Mary Hester, and Jason Zurawski. The science dmz: A network design pattern for data-intensive science. *2013 SC - International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 1–10, 2013.
- [9] M. Mellia, R. Lo Cigno, and F. Neri. Measuring ip and tcp behavior on edge nodes with tstat. *Comput. Netw.*, 47(1):1–21, January 2005.