Exploring the Impacts of Multiple I/O Metrics in Identifying I/O Bottlenecks

Izzet Yildirim
Illinois Institute of Technology
Chicago, IL, USA
iyildirim@hawk.iit.edu

Hariharan Devarajan
Lawrence Livermore National Laboratory
Livermore, CA, USA
hariharandev1@llnl.gov

Anthony Kougkas
Illinois Institute of Technology
Chicago, IL, USA
akougkas@iit.edu

Xian-He Sun
Illinois Institute of Technology
Chicago, IL, USA
sun@iit.edu

Kathryn Mohror
Lawrence Livermore National Laboratory
Livermore, CA, USA
kathryn@llnl.gov

ABSTRACT

HPC systems, driven by the rise of workloads with significant data requirements, face challenges in I/O performance. To address this, a thorough I/O analysis is crucial to identify potential bottlenecks. However, the multitude of metrics makes it difficult to pinpoint the causes of low I/O performance. In this work, we analyze three scientific workloads using three widely accepted I/O metrics. We demonstrate that different metrics uncover different I/O bottlenecks, highlighting the importance of considering multiple metrics for comprehensive I/O analysis.

1 INTRODUCTION

The escalating complexity of HPC systems and applications, particularly fueled by the rise of AI and ML workloads with their substantial data requirements, has resulted in significant challenges for I/O performance. The surge in data-driven tasks has placed immense pressure on the I/O subsystems, leading to performance issues that impede overall system efficiency. Addressing these challenges requires a thorough I/O analysis to uncover potential I/O bottlenecks. However, identifying these bottlenecks is a daunting task due to the multitude of metrics that must be considered. Metrics such as I/O bandwidth, transfer size, metadata operation rates, and various system-level factors all play critical roles in the overall I/O performance.

Some of the earliest studies [2, 10] concentrated on analyzing access patterns and transfer sizes to identify I/O performance issues. These studies highlighted significant variations in access patterns and transfer size distributions, prompting the need to consider additional factors to accurately assess I/O performance issues. As HPC systems expanded in scale and sophistication, their I/O operations became more intricate and challenging to analyze manually. I/O characterization tools (e.g. Darshan [1]) emerged as instrumental solutions that could automatically record detailed information about the I/O behavior of applications. This allowed studies like UMAMI [7], TOKIO [6], and IOMiner [12] to conduct large-scale I/O analysis using multiple metrics, such as I/O bandwidth and metadata operation rates. The findings from these studies demonstrate that the causes of low I/O performance in applications can be diverse. Therefore, it becomes imperative to consider multiple metrics simultaneously in a comprehensive I/O analysis to effectively capture the complex behaviors influencing overall I/O performance. For instance, while a particular application may exhibit good I/O bandwidth, it could still experience performance problems due to high metadata operation rates or suboptimal transfer sizes.

In this work, we present a methodology that leverages application I/O traces and a collection of I/O metrics to investigate the potential benefits of using multiple metrics in identifying I/O performance issues. By using multiple metrics simultaneously, this methodology distinguishes itself by offering a comprehensive understanding of the interconnected factors that influence I/O performance. We analyze three scientific workloads, each exhibiting diverse I/O behaviors. Using widely accepted I/O performance metrics, namely I/O time, I/O bandwidth, and I/O operation per second (IOPS), we evaluate the I/O performance of these workloads. Through our evaluations, we successfully identify various performance issues using different metrics. Moreover, we observe that certain metrics are better suited to capture particular I/O behaviors. Our key findings can be summarized as follows:

(1) Different metrics uncover different I/O bottlenecks.
(2) Specific I/O behaviors can only be captured by certain metrics.

2 IMPACTS OF MULTIPLE I/O METRICS

2.1 Methodology

We chose three scientific workloads with diverse I/O behaviors: CM1 [9], HACC [3], and Montage [5]. For this work, we utilized three time-based performance metrics to analyze the I/O performance of the scientific workloads: I/O time, I/O bandwidth, and IOPS. I/O time measures the time taken to perform I/O operations on a storage system. I/O bandwidth measures the total amount of data that can be read or written per second. IOPS represents the number of read and write operations a storage system can perform in a given second.

To detect I/O bottlenecks, we utilized distinct criteria tailored to each performance metric. For I/O time, we marked records where the time exceeded 90% of the maximum I/O time observed per process as I/O bottlenecks. For I/O bandwidth, we marked records where the throughput was below 10MB/s as I/O bottlenecks. Finally, for IOPS, we marked records where the I/O operation rates were less than 10% of the maximum IOPS observed per process as I/O bottlenecks.

2.2 Evaluation

We run the experiments on the Lassen supercomputer at Lawrence Livermore National Laboratory (LLNL) [4]. To capture I/O traces with the required level of granularity for this work, we utilized Recorder [11]. We used the Pandas [8] library as our analysis tool.

2.2.1 HACC. HACC is a cosmology workload that simulates the universe's evolution using particle-mesh techniques. It

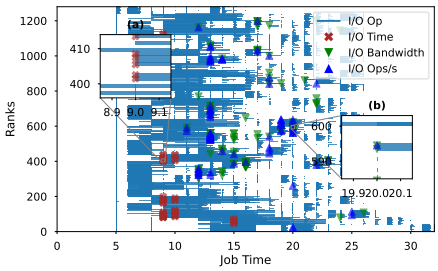

Figure 1: (a) Metadata contention due to GPFS causes 90% I/O time. (b) High parallelism during checkpointing result in low I/O bandwidth and IOPS.

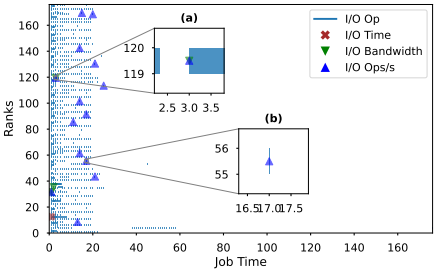

Figure 2: (a) Small "read"s on FITS files lead to very low I/O bandwidth and IOPS. (b) Slow "open"s during image generation result in low IOPS.

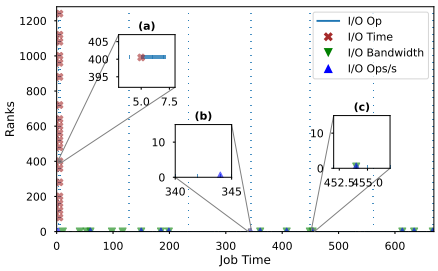

Figure 3: (a) Metadata contention causes 90% I/O time. (b) Metadata dominance leads to low IOPS. (c) Small "write"s result in low I/O bandwidth and IOPS.

includes an isolated I/O kernel called HACCIO, representing typical I/O workload in scientific simulations. The application uses 16M particles as input, writes nine variables, and performs checkpointing and restart.

The evaluation results are presented in Figure 1, where the x-axis represents the job time in seconds and the y-axis represents the ranks. The observations are twofold: (1) When multiple ranks attempt to "open" simulation files concurrently on GPFS, it results in contention within the parallel file system. This contention, resulting in more than 90% of the I/O time per process being consumed by metadata operations, leads to I/O bottlenecks per I/O time [Figure 1(a)]. (2) High parallelism during checkpointing also leads to contention on GPFS, resulting in I/O bottlenecks per both I/O BW and IOPS due to very low I/O bandwidth and IOPS [Figure 1(b)].

The findings demonstrate that the contention on GPFS can only be detected using the I/O time metric and not through I/O bandwidth or IOPS metrics. The reason behind this is that the latter two metrics only account for data operations and do not take metadata operations into consideration. Thus, it becomes evident that certain behaviors can only be identified by employing specific metrics.

2.2.2 Montage. Montage is a mosaic engine that converts sky-survey data from FITS files to PNG images. The collection of FITS images is divided into multiple segments, and these segments are processed in parallel to create PNG images. Specifically, 1024 FITS files are distributed among 32 nodes, with each node handling one segment containing 16 FITS files.

The evaluation results are presented in Figure 2, where the x-axis represents the job time in seconds and the y-axis represents the ranks. The observations are twofold: (1) Small "read"s (<3KB) on FITS files during initialization leads to I/O bottlenecks per both I/O bandwidth and IOPS. (2) Slow "open"s during PNG image generation causes I/O bottlenecks per IOPS.

The findings reveal that although small I/O operations, such as small "read"s, do not consume a significant amount of the maximum I/O time per process, they can still be detected as bottlenecks due to very low I/O bandwidth or IOPS.

2.2.3 CM1. CM1 is an atmospheric-simulation used to model thunderstorms and tornadoes. The simulation has separate read, write, and compute phases. The application uses configuration files of size 16MB to generate data and produces more than 750 files for different simulation steps. Each step generates files totaling approximately 128MB in size.

The evaluation results are presented in Figure 3, where the x-axis represents the job time in seconds and the y-axis represents the ranks. The observations are threefold: (1) During initialization, each first rank per node simultaneously "open"s the same configuration file, causing metadata contention. As a result, the application spends over 90% of its I/O time in this phase, leading to the detection of I/O bottlenecks per I/O time [Figure 3(a)]. (2) Simulation data writes are dominated by metadata operations, accounting for more than 80% of total I/O operations. Consequently, this leads to very low IOPS and and is detected as I/O bottlenecks per IOPS [Figure 3(b)]. (3) Simulation data writes dominated by small "write"s exhibit very low I/O BW and IOPS, hence are detected as I/O bottlenecks per both I/O BW and IOPS [Figure 3(c)].

The findings showcase that capturing I/O bottlenecks during the application's write phase (by rank 0) is only possible by considering I/O bandwidth or IOPS metrics. This is because certain I/O operations consume less than 20% of the maximum I/O time per process, making them undetectable as bottlenecks by the I/O time metric. However, these same operations exhibit very low I/O bandwidth or IOPS, making them noticeable as bottlenecks through those metrics. Moreover, it is shown that although there may be overlapping I/O bottlenecks based on I/O bandwidth and IOPS, this is not always the case. Therefore, each metric should be independently considered.

2.3 Conclusion

In this work, we presented a comprehensive I/O analysis using multiple metrics, namely I/O time, I/O bandwidth, and IOPS. Through the evaluation of three diverse scientific workloads, we demonstrated that different metrics uncover different I/O bottlenecks. Our findings demonstrate that specific I/O behaviors, such as contention on GPFS, can only be identified through certain metrics, further highlighting the need for considering multiple metrics.

ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under the DOE Early Career Research Program (LLNL-ABS-854292). Also, the material is based upon work supported by the National Science Foundation under Grant no. NSF OAC-2104013, OCI-1835764, CSR-1814872, and CSSI-2104013.

REFERENCES

[1] Philip Carns, Robert Latham, Robert Ross, Kamil Iskra, Samuel Lang, and Katherine Riley. 2009. 24/7 Characterization of petascale I/O workloads. In *2009 IEEE International Conference on Cluster Computing and Workshops*. IEEE, New Orleans, LA, USA, 1–10. https://doi.org/10.1109/CLUSTR.2009.5289150

[2] Phyllis E. Crandall, Ruth A. Aydt, Andrew A. Chien, and Daniel A. Reed. 1995. Input/output characteristics of scalable parallel applications. In *Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM) - Supercomputing '95*. ACM Press, San Diego, California, United States, 59–es. https://doi.org/10.1145/224170.224396

[3] Katrin Heitmann, Thomas D. Uram, Hal Finkel, Nicholas Frontiere, Salman Habib, Adrian Pope, Esteban Rangel, Joseph Hollowed, Danila Korytov, Patricia Larsen, Benjamin S. Allen, Kyle Chard, and Ian Foster. 2019. HACC Cosmological Simulations: First Data Release. *The Astrophysical Journal Supplement Series* 244, 1 (Sept. 2019), 17. https://doi.org/10.3847/1538-4365/ab3724

[4] HPC @ LLNL. 2023. Lassen. https://hpc.llnl.gov/hardware/compute-platforms/lassen

[5] Joseph C. Jacob, Daniel S. Katz, G. Bruce Berriman, John C. Good, Anastasia C. Laity, Ewa Deelman, Carl Kesselman, Gurmeet Singh, Mei Hui Su, Thomas A. Prince, and Roy Williams. 2009. Montage: a grid portal and software toolkit for science-grade astronomical image mosaicking. *International Journal of Computational Science and Engineering* 4, 2 (2009), 73. https://doi.org/10.1504/IJCSE.2009.026999

[6] Glenn K Lockwood, Nicholas J Wright, Shane Snyder, Philip Carns, George Brown, and Kevin Harms. 2018. TOKIO on ClusterStor: Connecting Standard Tools to Enable Holistic I/O Performance Analysis. *Proceedings of the 2018 Cray User Group* (2018).

[7] Glenn K. Lockwood, Wucherl Yoo, Suren Byna, Nicholas J. Wright, Shane Snyder, Kevin Harms, Zachary Nault, and Philip Carns. 2017. UMAMI: a recipe for generating meaningful metrics through holistic I/O performance analysis. In *Proceedings of the 2nd Joint International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems - PDSW-DISCS '17*. ACM Press, Denver, Colorado, 55–60. https://doi.org/10.1145/3149393.3149395

[8] Open-source. 2008. Pandas. https://pandas.pydata.org/

[9] Hafizur Rahman, Michel M. Verstraete, and Bernard Pinty. 1993. Coupled surface-atmosphere reflectance (CSAR) model: 1. Model description and inversion on synthetic data. *Journal of Geophysical Research* 98, D11 (1993), 20779. https://doi.org/10.1029/93JD02071

[10] A. L. Narasimha Reddy and Prithviraj Banerjee. 1990. A study of I/O behavior of perfect benchmarks on a multiprocessor. *ACM SIGARCH Computer Architecture News* 18, 2SI (June 1990), 312–321. https://doi.org/10.1145/325096.325157

[11] Chen Wang, Jinghan Sun, Marc Snir, Kathryn Mohror, and Elsa Gonsiorowski. 2020. Recorder 2.0: Efficient Parallel I/O Tracing and Analysis. In *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, New Orleans, LA, USA, 1–8. https://doi.org/10.1109/IPDPSW50202.2020.00176

[12] Teng Wang, Shane Snyder, Glenn Lockwood, Philip Carns, Nicholas Wright, and Suren Byna. 2018. IOMiner: Large-Scale Analytics Framework for Gaining Knowledge from I/O Logs. In *2018 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, Belfast, 466–476. https://doi.org/10.1109/CLUSTER.2018.00062