UMAMI: A Recipe for Generating Meaningful Metrics through Holistic I/O Performance Analysis

Glenn K. Lockwood, Wucherl Yoo,
Suren Byna, Nicholas J. Wright
glock@lbl.gov
Lawrence Berkeley National Laboratory

Shane Snyder, Kevin Harms,
Zachary Nault, Philip Carns
Argonne National Laboratory

ABSTRACT

I/O efficiency is essential to productivity in scientific computing, especially as many scientific domains become more data-intensive. Many characterization tools have been used to elucidate specific aspects of parallel I/O performance, but analyzing components of complex I/O subsystems in isolation fails to provide insight into critical questions: how do the I/O components interact, what are reasonable expectations for application performance, and what are the underlying causes of I/O performance problems? To address these questions while capitalizing on existing component-level characterization tools, we propose an approach that combines on-demand, modular synthesis of I/O characterization data into a unified monitoring and metrics interface (UMAMI) to provide a normalized, holistic view of I/O behavior.

We evaluate the feasibility of this approach by applying it to a month-long benchmarking study on two distinct large-scale computing platforms. We present three case studies that highlight the importance of analyzing application I/O performance in context with both contemporaneous and historical component metrics, and we provide new insights into the factors affecting I/O performance. By demonstrating the generality of our approach, we lay the groundwork for a production-grade framework for holistic I/O analysis.

ACM Reference Format:
Glenn K. Lockwood, Wucherl Yoo, Suren Byna, Nicholas J. Wright and Shane Snyder, Kevin Harms, Zachary Nault, Philip Carns. 2017. UMAMI: A Recipe for Generating Meaningful Metrics through Holistic I/O Performance Analysis. In *PDSW-DISCS'17: PDSW-DISCS'17: Second Joint International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems, November 12–17, 2017, Denver, CO, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3149393.3149395

ACM acknowledges that this contribution was authored or co-authored by an employee, contractor, or affiliate of the United States government. As such, the United States government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for government purposes only.

PDSW-DISCS'17, November 12–17, 2017, Denver, CO, USA

© 2017 Association for Computing Machinery.

ACM ISBN 978-1-4503-5134-8/17/11…$15.00
https://doi.org/10.1145/3149393.3149395

1 INTRODUCTION

The stratification of performance and capacity in storage technology is resulting in increasingly complex parallel storage system architectures. Leadership-class systems are now being deployed with flash-based burst buffers [9] that provide even higher performance than disk-based file systems do [2]. This performance comes at the cost of increasing complexity, however, making I/O performance analysis increasingly difficult.

The current practice is to monitor each I/O component separately, often resulting in telemetric data that are not directly compatible. For example, server-side monitoring tools such as LMT [7] continuously measure a few metrics over time, while application-level profiling tools such as Darshan [5] record extended metrics for a single job. At present, the gaps of information resulting from these incompatibilities are filled by expert institutional knowledge and intuition.

Absent this expert knowledge, it is challenging to determine whether a job's I/O performance is normal on any specific HPC system given the application's I/O pattern. Relying on intuition to tie together different I/O data sources and decide whether a job performed as expected becomes unsustainable as storage subsystems become more complex. Thus, it is becoming critical to integrate component-level data and present a coherent, holistic view of the I/O subsystem and its interdependent behavior.

To demonstrate the benefits of a holistic approach, we have conducted a month-long benchmarking study of several I/O-intensive applications on two architecturally distinct production HPC systems. Using component-level data already being collected on those systems, we analyze data integrated from application-level profiling, file system servers, and other system-level components. By showing how these metrics vary over the one-month experiment in a unified monitoring and metrics interface (UMAMI), this holistic approach is able to differentiate general performance expectations for different I/O motifs (analogous to the climate of the I/O system) from transient effects (analogous to the weather of the I/O system). We use this notion of the *I/O climate* to encompass the characteristics of storage components, their age

and capacity, and the way they respond to a specific workload. Complementary to the I/O climate, the *I/O weather* is determined by the transitory state of the job scheduler load, I/O contention, and short-term failure events.

The primary contributions of this work are as follows.

- We demonstrate that the degree of variability in I/O performance is a function of the storage system architecture, the application's I/O motif, and the overall file system climate. Different I/O patterns expose different degrees of performance variation on different parallel file system architectures, and the nature of a file system's typical workload also shapes performance variability.
- We show that I/O performance is affected by both intrinsic application characteristics and extrinsic storage system factors. Contention with other I/O workloads for storage system bandwidth is not the only factor that affects performance, and we highlight cases where metadata contention and file system fullness dramatically impact performance.
- We show that no single monitoring metric predicts I/O performance universally across HPC platforms. The system architecture, configuration parameters, workload characteristics, and system health all play varying roles.

2 EXPERIMENTAL METHODS

To examine the utility and generality of integrating data from multiple component-level monitoring tools into a single view (UMAMI), we conducted an I/O benchmark study on two distinct HPC platforms described in Table 1.

2.1 NERSC Edison

Edison is a Cray XC-30 system at the National Energy Research Scientific Computing Center (NERSC). Its scratch1 and scratch2 Lustre file systems are identically configured, and users are evenly distributed across them. However, access to Edison's scratch3 file system is granted only to users who require high parallel bandwidth, and therefore the scratch3 file system should reflect larger, more coherent I/O traffic.

Edison's architecture routes I/O traffic from its Aries high-speed network to the InfiniBand SAN fabric via LNET I/O nodes. Routing is configured such that each LNET I/O node handles traffic for only one of the three Edison file systems to ensure that each file system's traffic is isolated as it transits I/O nodes. This also allows jobs of any size to use the maximum number of I/O nodes for each file system. In this work,

Table 1: Description of test platforms

	Platform	FS Name (Type)	# ION, SVR,LUN	Size	Peak Rate
Edison NERSC	Cray XC 5,586 CN	scratch1 (Lustre)	9,24,24	2.2 PB	48 GB/s
		scratch2 (Lustre)	9,24,24	2.2 PB	48 GB/s
		scratch3 (Lustre)	13,36,36	3.3 PB	72 GB/s
Mira ALCF	IBM BG 49,152 CN	mira-fs1 (GPFS)	384,48,336	7.0 PB	90 GB/s

Table 2: Benchmark configuration parameters

Application	I/O Motif	Mira Size	Edison Size
HACC	POSIX file per proc	1.5 TiB	2.0 TiB
VPIC / BD-CATS	HDF5 shared file	1.0 TiB	2.0 TiB
IOR	POSIX file per proc	1.0 TiB	2.0 TiB
IOR	MPI-IO shared file	1.0 TiB	0.5 TiB

all output data was striped over all OSTs in each file system, and the input parameters listed in Table 2 were chosen to saturate each file system's bandwidth. Our IOR benchmarks demonstrated 90% of the theoretical maximum performance.

2.2 ALCF Mira

Mira is an IBM Blue Gene/Q system at the Argonne Leadership Computing Facility (ALCF). In addition to the Spectrum Scale (GPFS) servers and LUNs listed, six of the network shared disk (NSD) servers also serve metadata from SSD-based LUNs. Jobs on Mira are allocated I/O nodes and compute nodes in a fixed ratio (1 I/O node for every 128 compute node), causing storage bandwidth to scale linearly with the size of the job. To keep compute resource consumption low, we opted to run every benchmark using 1,024 compute nodes, giving them eight I/O nodes and an aggregate peak bandwidth of ~25 GB/sec. Our IOR configuration achieved 80% of peak performance for this job size.

2.3 I/O performance regression tests

For this study, we ran the following benchmark applications using three file systems across 39 days on Edison (1,014 benchmark runs) and on one file system across 29 days on Mira (118 benchmark runs).

- **Hardware Accelerated Cosmology Code (HACC)**, a cosmology application [8], configured to generate 96 MiB per process using POSIX file-per-process checkpoint I/O.
- **Vector Particle-In-Cell (VPIC)**, a plasma physics application [3], configured to write 1.0 GiB per process to a single HDF5 file using the H5Part API [1], and **BD-CATS**, a clustering analysis system used to analyze VPIC output data [14], configured to read 75% of our VPIC output to emulate a 3D clustering analysis.
- **IOR**, a widely used tool to characterize parallel file system performance [12, 16–18], used to determine each file system's performance under optimal I/O workloads.

All benchmarks were run using 1,024 and 128 nodes (16 processes per node) on Mira and Edison, respectively. These scales were chosen to sufficiently utilize the capability of the storage system while limiting the core-hour consumption, and the resulting data volumes are summarized in Table 2.

3 DATA SOURCES

We drew a total of 58 different metrics from a variety of monitoring tools already in production use on Mira and Edison over the course of our study.

3.1 Application behavior

To capture the I/O patterns and user-observable application performance in this study, we used the Darshan I/O characterization tool [5] which transparently records statistics about an application's I/O behavior at runtime. It imposes minimal overhead because it defers the reduction of these statistics until the application exits, allowing it to be deployed for all production applications on large-scale systems without perturbing performance. Both Mira and Edison link Darshan into all compiled applications by default.

3.2 Storage system traffic

Storage system traffic monitoring provides aggregate systemwide I/O workload metrics such as bytes read/written and operation counts for reads, writes, and metadata. These time-series data are collected with minimal impact on application performance because they are gathered on the storage servers, not compute nodes. On both Mira and Edison, these metrics were collected at five-second intervals using file-system-specific tools.

Lustre Monitoring Tools (LMT) aggregates Lustre-specific counters on each object storage server (OSS) and metadata server (MDS) and presents them via a MySQL database. LMT provides data including bytes read/written, CPU load averages, and metadata operation rates. *ggiostat* is a tool developed at the ALCF to collect similar data from IBM Spectrum Scale file systems. It retrieves and stores metrics from server and client clusters and provides bytes read/written and operation counts for reads, writes, and metadata operations.

3.3 Health monitoring

Health-monitoring data describe what components are offline, failed-over, or in another degraded state and how much free capacity remains on the available devices. On Edison, the fullness of each Lustre object storage target (OST) is recorded every fifteen minutes. On Mira, the fullness of each LUN and the failure status of each server is recorded upon job submission. The mapping between OSTs/LUNs and OSS/NSD servers are also logged to identify degraded devices.

3.4 Job scheduling and topology

Job-scheduling data provides details on the jobs that are concurrently running on a system and can help identify cases where I/O contention results from competing jobs. In this study, we tracked the number of other jobs that were running and the number of core-hours consumed systemwide during the time our benchmark jobs ran.

To identify any effects of job placement on Edison and Mira's high-speed networks, we calculate a job's maximum radius as an approximation of that job's degree of delocalization. Using the topological coordinates of each job's compute node allocation to derive a center of mass of a job, we define this metric as the maximum distance between that center of mass and a compute node.

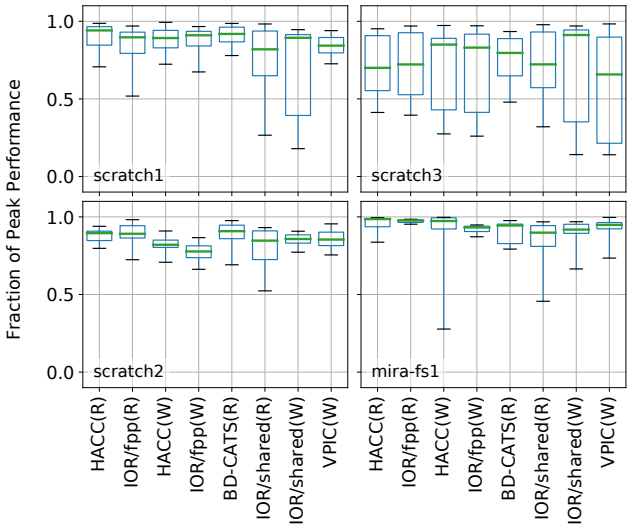

Figure 1: I/O performance grouped by test applications and read(R)/write(W) mode. Whiskers represent the 5th and 95th percentiles.

4 BASELINE I/O PERFORMANCE

Because variation in peak I/O performance is caused by different I/O access patterns [12, 16, 17], we first establish the baseline variation of each benchmark on each system. We define the *fraction of peak performance* as the observed I/O bandwidth (performance) of a job divided by the maximum performance observed for all jobs *of the same I/O motif* as listed in Table 2 and whether the job read or wrote.

The distribution of fraction peak performance (Figure 1) reveals that the degree of variation *within* each application varies with each file system. For example, the HACC write workload is susceptible to a long tail of performance loss on mira-fs1 despite mira-fs1's overall lower variation as evidenced by the distance between all I/O motifs' whiskers relative to the Edison file systems. Edison's scratch3 also shows broad performance variation for VPIC, contrasting with the narrower variation of VPIC on other systems. We conclude that such variability results from factors intrinsic to both the application and the file system; different I/O motifs result in different levels of performance *and* variability.

Furthermore, Figure 1 shows that variation is not only a function of the file system architecture; all Edison file systems are Lustre-based, yet Figure 1 shows a marked difference in variability between scratch1/scratch2 and scratch3. Thus, these differences in performance variation must be a function of differences in three factors of the I/O subsystem: (1) hardware architecture, evident when comparing the

distributions of mira-fs1 performance to Edison; (2) application I/O patterns, evident from the variation within any single file system; and (3) overall file system climate, evident by comparing the architecturally equivalent Edison scratch1/scratch2 with scratch3 file systems.

This finding underscores the importance of examining multiple sources of I/O characterization data in concert and with historical context to develop a full understanding of I/O performance.

5 INTEGRATED ANALYSIS

With an understanding of the baseline performance variation on each system and application, we then use the metrics described in Section 3 to analyze how extrinsic factors affect performance. Bandwidth contention from other jobs is an intuitive source of performance variation, so we define the *bandwidth coverage factor* (CF_{bw}) of a job j to quantify the effects of competing I/O traffic:

$$CF_{bw}(j) = \frac{N_{\text{bytes}}^{\text{Darshan}}(j)}{\sum_{t,s}^{\text{time,servers}} \left[N_{\text{bytes}}^{\text{LMT,ggiostat}}(t,s) \right]}, \quad (1)$$

where $N_{\text{bytes}}^{\text{Darshan}}$ are the bytes read and written by job j according to its Darshan log and $N_{\text{bytes}}^{\text{LMT,ggiostat}}$ are the bytes read and written to a file system server s during a 5-second time interval t. The time interval over which the job ran (*time*) and the servers to which the job wrote (*servers*) are also stored in the job's Darshan log [15]. CF_{bw} is a direct reflection of how much I/O traffic a job competed against in the underlying file systems. Relatedly, we also define CF_{IOPS} of IOPS (derived from Darshan and ggiostat) and $CF_{nodehrs}$ of node-hours (derived from job-scheduling data).

CF, system health data, and job topology data let us contextualize performance anomalies and quantify where a job's I/O performance falls on the spectrum of normalcy relative to jobs with similar motifs. To concisely display this information and identify metrics that most likely contribute to abnormal performance, we propose a unified monitoring and metrics interface (UMAMI) diagram as demonstrated in Figure 2. UMAMI presents historic measurements (the I/O climate) alongside a box plot that summarizes each metric. These time series plots highlight a job of interest and define the I/O weather at the time that job ran.

Overlaying this weather on the climate (dashed lines in the box plots) shows how each metric compares with the distribution of weather conditions before, after, or surrounding the job of interest to enable rapid differentiation of rare events from long-term performance problems. In the remainder of this section we use UMAMI diagrams to identify different factors that do and do not contribute to I/O performance loss. Although we collected 58 metrics for each job, we chose only the most relevant metrics to include in each UMAMI diagram

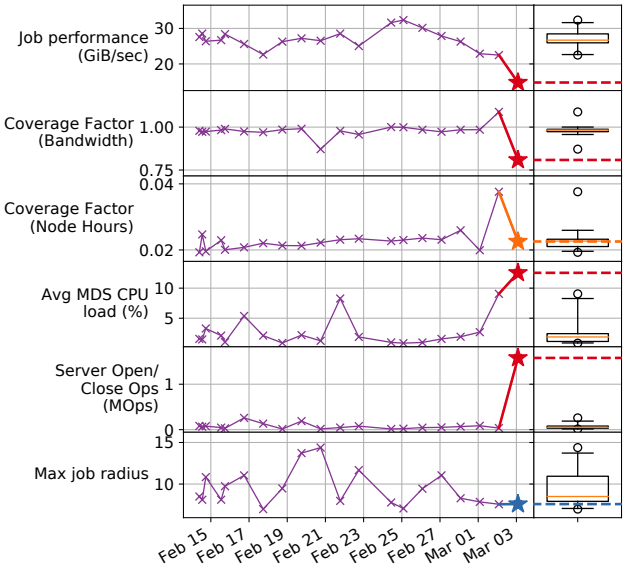

Figure 2: UMAMI of HACC write workloads on Edison scratch2. Left panes show measurements from other runs of the same motif; right panes show summaries. Stars highlight the job of interest and are colored red, orange, or blue if they are above, within, or below the overall quartiles. Whiskers indicate 5th and 95th percentiles; circles are outliers.

based on (1) which metrics most strongly correlated with performance, and (2) which metrics we expected to affect performance but did not.

These case studies were performed on Mira and Edison, but UMAMI's modularity allow it to analyze a variety of data sources and enables portable deployment for production use.

5.1 Case study: I/O contention

The UMAMI example in Figure 2 represents a HACC write test whose job performance measurement relative to previous HACC write jobs indicate statistically abnormal performance. This poor performance was accompanied by an unusually low CF_{bw} and high metadata load, highlighted as red dashed lines in the box plots that denote their place in the least-favorable quartile of past measurements.

Conversely, the maximum job radius fell into the most favorable quartile (indicated by the blue dashed line), and the number of concurrently running jobs ($CF_{nodehrs}$) was not abnormally large. Because normal performance was often observed even in cases when both of these metrics were abnormally poor, we conclude that the maximum job radius and $CF_{nodehrs}$ metrics are too coarse-grained to indicate poor performance, and more insight into what resources the competing jobs were actually consuming when each HACC job ran are required. Given this body of information, we attribute

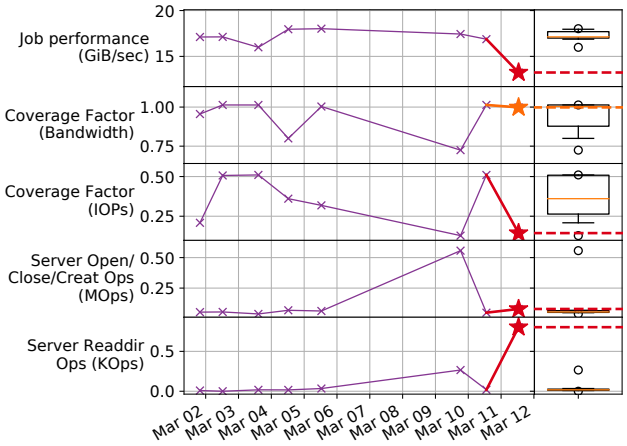

Figure 3: UMAMI demonstrating the climate surrounding VPIC write workloads on Mira compared with a most recent run, which showed highly unusual weather in the form of an excess of `readdir(3)` calls.

this HACC job's poor performance to I/O loads from other jobs that competed for both bandwidth and metadata rates.

5.2 Case study: metadata load

Figure 3 shows the UMAMI diagram for a poorly performing VPIC workload. CF_{bw} is within normal parameters, indicating normal (minimal) levels of bandwidth contention. CF_{IOPS} is abnormally low, although previous values have been equally low despite a lack of dramatic performance loss (e.g., on March 1 and March 9). The only metric that shows a unique, undesirable value is the number of readdir operations handled by the file system, indicating a file system traversal was running concurrently with this VPIC job. From this we infer that metadata load, not bandwidth contention, contributed to poor VPIC performance on March 11.

5.3 Case study: storage capacity

This holistic approach can also identify long-term performance degradation. Figure 4 shows the UMAMI of HACC on Edison scratch3 when *CF*s were not unusual despite an ongoing 2× slowdown over the normal 50 GiB/sec between February 24 and March 9. The magnitude of performance loss closely followed the maximum CPU load on the Lustre OSSes, and this period also coincided with scratch3 OSTs approaching 100% fullness. The relationship between CPU load, OST fullness, and I/O performance points to an increasing cost of scavenging empty blocks on writes, and this behavior is consistent with known performance losses that result from Lustre OSTs filling [13]. Furthermore, I/O performance was restored on March 9 which is when NERSC staff initiated a file system purge. Thus, we conclude that this long-term performance degradation was the result of poor file system health resulting from Edison scratch3 being critically full.

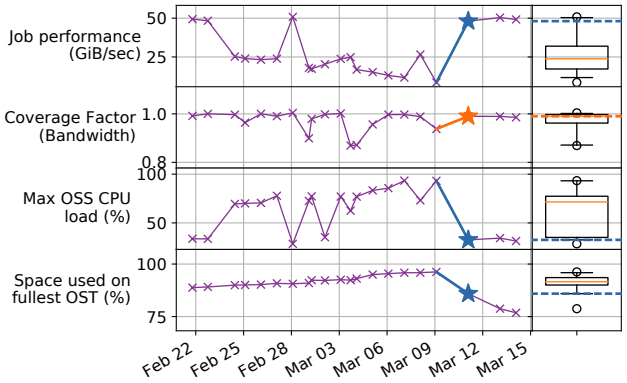

Figure 4: UMAMI of HACC write performance on Edison's scratch3 file system showing a longer-term period of performance degradation that was associated with poor file system health.

6 RELATED WORK

Several studies have explored how to combine and analyze multiple sources of I/O monitoring information. Kunkel et al. developed SIOX [10] to aggregate information from multiple components of the I/O stack, but its reliance on instrumented versions of application libraries to collect metrics makes production deployment challenging. Liu et al. developed AID [11] to perform detailed analysis to server-side I/O logs in order to deduce application-level I/O patterns and make scheduling recommendations. The clustering approach implemented by AID may be able to automatically identify I/O motifs suitable for generating UMAMI diagrams.

Other studies have explored how to quantify and combat various types of I/O performance variation. Lofstead et al. observed that variability is caused by both external and internal interference within an application [12], and they proposed an adaptive strategy that coordinates I/O activity within an application. Similarly, Dorier et al. proposed middleware for coordinating I/O across applications to manage external interference [6]. Yildiz et al.'s study of I/O interference in a testbed environment found that poor flow control in the I/O path [18] also contributes to variation, indicating that network-related metrics would be a valuable addition to UMAMI diagrams. Carns et al. reported I/O variability for seven common production jobs during a two-month study [4] and, similar to our findings, suggested that some access patterns are more susceptible to variability.

7 CONCLUSIONS

By integrating data captured with existing tools from applications, storage systems, system health, and job scheduling, we demonstrated that holistically examining all components of the I/O subsystem is essential for understanding I/O performance variation. We performed a month-long benchmarking

study and characterized the I/O *climate* on each system, then presented several case studies to demonstrate instances of abnormal I/O *weather* and their effects on I/O performance.

Integrating metrics into the UMAMI diagram revealed that contention with other workloads for bandwidth, metadata op rates, and storage capacity can, but do not always, impact performance. No single metric predicts I/O performance universally; the most significant metrics depend on systems' architecture, configuration, workload characteristics, and health, while factors such as job radius and $CF_{nodehrs}$ do not capture enough detail to indicate performance loss. These findings provide a basis for improving monitoring tools to capture more detailed metrics that can better predict I/O performance.

ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, under contracts DE-AC02-05CH11231 and DE-AC02-06CH11357 (Project: A Framework for Holistic I/O Workload Characterization, Program manager: Dr. Lucy Nowell). This research used resources and data generated from resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231 and the Argonne Leadership Computing Facility, a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

REFERENCES

[1] A. Adelmann, A. Gsell, B. Oswald, T. Schietinger, W. Bethel, J. M. Shalf, C. Siegerist, and K. Stockinger. 2007. Progress on H5Part: a portable high performance parallel data interface for electromagnetics simulations. In *2007 IEEE Particle Accelerator Conference (PAC)*. 3396–3398.

[2] Wahid Bhimji, Debbie Bard, Melissa Romanus, David Paul, Andrey Ovsyannikov, Brian Friesen, Matt Bryson, Joaquin Correa, Glenn K. Lockwood, Vakho Tsulaia, Surendra Byna, Steve Farrell, Doga Gursoy, Chris Daley, Vince Beckner, Brian Van Straalen, David Trebotich, Craig Tull, Gunther Weber, Nicholas J. Wright, Katie Antypas, and Prabhat. 2016. Accelerating Science with the NERSC Burst Buffer Early User Program. In *Proceedings of the 2016 Cray User Group*. London. https://www.nersc.gov/assets/Uploads/Nersc-BB-EUP-CUG.pdf

[3] K. J. Bowers, B. J. Albright, L. Yin, B. Bergen, and T. J T Kwan. 2008. Ultrahigh performance three-dimensional electromagnetic relativistic kinetic plasma simulation. *Physics of Plasmas* 15, 5 (may 2008), 55703.

[4] Philip Carns, Kevin Harms, William Allcock, Charles Bacon, Samuel Lang, Robert Latham, and Robert Ross. 2011. Understanding and improving computational science storage access through continuous characterization. *ACM Transactions on Storage (TOS)* 7, 3 (2011), 8.

[5] Philip Carns, Robert Latham, Robert Ross, Kamil Iskra, Samuel Lang, and Katherine Riley. 2009. 24/7 characterization of petascale I/O workloads. In *Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER'09)*. IEEE, 1–10.

[6] Matthieu Dorier, Gabriel Antoniu, Rob Ross, Dries Kimpe, and Shadi Ibrahim. 2014. CALCioM: Mitigating I/O interference in HPC systems through cross-application coordination. In *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*. IEEE, 155–164.

[7] Jim Garlick and Christopher Morrone. 2010. Lustre Monitoring Tools. (2010). https://github.com/LLNL/lmt

[8] Salman Habib, Vitali A. Morozov, Hal Finkel, Adrian Pope, Katrin Heitmann, Kalyan Kumaran, Tom Peterka, Joseph A. Insley, David Daniel, Patricia K. Fasel, Nicholas Frontiere, and Zarija Lukic. 2012. The Universe at Extreme Scale: Multi-Petaflop Sky Simulation on the BG/Q. *CoRR* abs/1211.4864 (2012). http://arxiv.org/abs/1211.4864

[9] Dave Henseler, Benjamin Landsteiner, Doug Petesch, Cornell Wright, and Nicholas J Wright. 2016. Architecture and Design of Cray DataWarp. In *Proceedings of the 2016 Cray User Group*. London. https://cug.org/proceedings/cug2016_proceedings/includes/files/pap105.pdf

[10] Julian M. Kunkel, Michaela Zimmer, Nathanael Hübbe, Alvaro Aguilera, Holger Mickler, Xuan Wang, Andriy Chut, Thomas Bönisch, Jakob Lüttgau, Roman Michel, and Johann Weging. 2014. The SIOX Architecture — Coupling Automatic Monitoring and Optimization of Parallel I/O. In *Proceedings of the 29th International Conference on Supercomputing - Volume 8488 (ISC 2014)*. Springer-Verlag New York, Inc., New York, NY, USA, 245–260.

[11] Yang Liu, Raghul Gunasekaran, Xiaosong Ma, and Sudharshan S Vazhkudai. 2016. Server-side Log Data Analytics for I/O Workload Characterization and Coordination on Large Shared Storage Systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'16)*. IEEE Press, 70:1—-70:11.

[12] Jay Lofstead, Fang Zheng, Qing Liu, Scott Klasky, Ron Oldfield, Todd Kordenbrock, Karsten Schwan, and Matthew Wolf. 2010. Managing Variability in the IO Performance of Petascale Storage Systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'10)*. IEEE, 1–12.

[13] Sarp Oral, James Simmons, Jason Hill, Dustin Leverman, Feiyi Wang, Matt Ezell, Ross Miller, Douglas Fuller, Raghul Gunasekaran, Youngjae Kim, et al. 2014. Best practices and lessons learned from deploying and operating large-scale data-centric parallel file systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Press, 217–228.

[14] Md. Mostofa Ali Patwary, Suren Byna, Nadathur Rajagopalan Satish, Narayanan Sundaram, Zarija Lukić, Vadim Roytershteyn, Michael J. Anderson, Yushu Yao, Prabhat, and Pradeep Dubey. 2015. BD-CATS: Big Data Clustering at Trillion Particle Scale. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'15)*. 6:1–6:12.

[15] Shane Snyder, Philip Carns, Kevin Harms, Robert Ross, Glenn K Lockwood, and Nicholas J Wright. 2016. Modular HPC I/O characterization with Darshan. In *Proceedings of the 5th Workshop on Extreme-Scale Programming Tools*. IEEE Press, 9–17.

[16] Andrew Uselton, Mark Howison, Nicholas J. Wright, David Skinner, Noel Keen, John Shalf, Karen L. Karavanic, and Leonid Oliker. 2010. Parallel I/O performance: From events to ensembles. In *Proceedings of the IEEE International Symposium on Parallel & Distributed Processing (IPDPS'10)*. IEEE, 1–11.

[17] Bing Xie, Jeffrey Chase, David Dillow, Oleg Drokin, Scott Klasky, Sarp Oral, and Norbert Podhorszki. 2012. Characterizing output bottlenecks in a supercomputer. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'12)*. IEEE, 1–11.

[18] Orcun Yildiz, Matthieu Dorier, Shadi Ibrahim, Rob Ross, and Gabriel Antoniu. 2016. On the Root Causes of Cross-Application I/O Interference in HPC Storage Systems. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 750–759.