ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Lemont, Illinois 60439

# Analysis and Correlation of Application I/O Performance and System-Wide I/O Activity[1]

**Sandeep Madireddy, Prasanna Balaprakash, Philip Carns, Robert Latham, Robert Ross, Shane Snyder, and Stefan M. Wild**

Mathematics and Computer Science Division

Preprint ANL/MCS-P7036-0417

April 2017

**Updates to this preprint may be found at**
http://www.mcs.anl.gov/publications

[1]This material was based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Contract No. DE-AC02-06CH11357.

# Analysis and Correlation of Application I/O Performance and System-Wide I/O Activity

Sandeep Madireddy, Prasanna Balaprakash, Philip Carns, Robert Latham,
Robert Ross, Shane Snyder, and Stefan M. Wild
Email: {smadireddy, pbalapra, carns, robl, rross, ssnyder, wild}@mcs.anl.gov
Mathematics and Computer Science Division
Argonne National Laboratory
9700 S. Cass Avenue
Lemont, IL 60439

***Abstract*—Storage resources in high-performance computing are shared across all user applications. Consequently, storage performance can vary markedly, depending not only on an application's workload but also on what other activity is concurrently running across the system. This variability in storage performance is directly reflected in overall execution time variability, thus confounding efforts to predict job performance for scheduling or capacity planning. I/O variability also complicates the seemingly straightforward process of performance measurement when evaluating application optimizations. In this work we present a methodology to measure I/O contention with more rigor than in prior work. We apply statistical techniques to gain insight from application-level statistics and storage-side logging. We examine different correlation metrics for relating system workload to job I/O performance and identify an effective and generally applicable metric for measuring job I/O performance. We further demonstrate that the system-wide monitoring granularity can directly affect the strength of correlation observed. Insufficient granularity and measurements can hide the correlations between application I/O performance and system-wide I/O activity.**

## I. Introduction

Performance variability, manifested as unpredictability in application execution time, is an impediment to efficient resource management and productivity in scientific computing [1]. I/O performance is one of the most prominent underlying contributors to this execution time variability. High-performance computing (HPC) storage resources are simultaneously shared by a large number of applications, and I/O behavior in those applications is often characterized by intense bursts of data access interleaved with intervals of computation. This mix of bursty, uncoordinated storage system traffic causes significant fluctuations in the I/O performance perceived by individual applications. The issue is exacerbated by growing complexity in I/O architectures that integrate more diverse and less-well-understood storage technologies in order to maximize the price/performance ratio.

The problem of I/O performance variability has motivated numerous research efforts with the objective of mitigating I/O contention. These efforts, which are summarized in Sec. II, often focus on application-specific I/O strategies or the nuances of specific computing platforms because no widely accepted, reusable methodology exists for characterizing variability across platforms and applications. This gap in our understanding makes it difficult to consistently detect variability, attribute the cause, and apply variability-aware optimizations across diverse environments.

To close this gap in understanding, we have developed a methodology, including both measurement techniques and analysis methods, to correlate system-wide I/O activity with job performance. We have applied this methodology to two months of production system data collected in a previous study, focusing on three scientific application groups. Here we make the following contributions:

- Systematic methodology to obtain associations between a job's I/O time and system-wide I/O;
- Job-level I/O performance metric that maximizes correlation accuracy while remaining generally applicable;
- Quantification of the impact of system-level sampling granularity on performance correlation.

We envision these contributions as important stepping stones toward more rigorous analysis of I/O performance and its variability that can be applied to improve scheduling strategies, dynamically adapt to anticipated workloads, enable more effective application-level optimizations, and improve our ability to measure and interpret performance. Our analysis also motivates future work in I/O instrumentation to better support these use cases.

The remainder of this paper is organized as follows. In Sec. III we provide formal definitions of several complementary empirical metrics for I/O time and system-wide I/O activity. We focus on three application groups; the characteristics of these applications are reviewed in Sec. IV. We define the statistical correlation metrics and our methodology in Sec. V. In Sec. VI we examine the application data by using this methodology, and we discuss the correlations observed and demonstrate the effect of measurement granularity on our results. Section. VII summarizes the conclusions and briefly discusses avenues for future work.

## II. Related Work

Previous studies have quantified I/O variability and its causes in specific HPC environments. Carns et al. reported I/O performance variability for seven frequently repeated production jobs during a two-month period [2]. Their results suggested that some access patterns are more susceptible to variability than others, but the root cause of the variability was not identified. Yildiz et al. performed a systematic study of I/O interference in a Grid'5000 testbed environment and found that significant variability often arose from poor flow control in the I/O path [3]. Uselton et al. applied statistical techniques to high-fidelity trace events in order to identify anomalous behavior including causes of I/O variance [4]. Recent studies have sought to correlate server-side performance logs with application performance, including metrics to help understand the I/O intensity and performance of various applications [5], [6].

One consequence of I/O variability is that it complicates the quantitative analysis of computer science innovations. Traeger et al. performed a multiyear study of storage benchmarking that highlighted frequent use of inadequate statistical methods when presenting storage research [7], a problem that is exacerbated on systems with more performance variance. Hoefler and Belli described additional challenges in interpreting computer system variability, including the observation that normal distributions are rarely seen in computer performance [8]. Studies that rely on automated methods (e.g., I/O autotuning [9], [10]) employ scalar values such as the minimum of many repetitions to set performance bounds in the presence of noisy data. Isaila et al. developed techniques to help address these challenges in autotuning and modeling by employing hybrid models that incorporate both analytical and black-box techniques [11].

Other researchers have explored strategies to mitigate I/O performance variability at the application level in order to improve user experience. Lofstead et al. observed that variability can arise not only from external interference but also from interference among the processes within an application [12]. They proposed an adaptive I/O strategy that coordinates I/O activity within an application according to the relative performance of file system servers. Dorier et al. proposed a middleware mechanism for coordinating I/O activity across applications to manage external interference [13]. They evaluated a collection of strategies that could be used to mitigate potential interference and evaluated their effectiveness. Isaila et al. introduced a hierarchy of controllers to enforce global, cross-layer coordination policies that minimize interference [14]. Son et al. evaluated the use of runtime probing to measure the performance of file system servers during application execution as part of a high-level I/O library [15]. This information was then used to select an optimal subset of storage servers to write data to.

Previous research in I/O variability has relied on case studies or architecture-specific features. Formalizing variability detection and analysis in a way that can be repeated over time and across platforms remains an open challenge.

## III. Data Acquisition and I/O Metrics

The data for this study was obtained on Intrepid, an IBM Blue Gene/P system deployed at the Argonne Leadership Computing Facility (ALCF) from 2007 to 2013. The data collection methods are described in [2], and an anonymized copy of the data is publicly available as part of the ALCF I/O data repository [16].

Application-side I/O information was obtained by using Darshan [2], a lightweight I/O characterization tool that was deployed on Intrepid. Among the data instrumented by Darshan are high-level job details and a compact set of counters, timers, and timestamps describing the I/O workload for each file accessed by the application. The total run time for a job ($\phi^J$) is easily calculated as the difference between the job start and end timestamps provided by Darshan. To quantify job I/O performance, Darshan decomposes general job I/O activity into activity involving shared files (files that were opened by all MPI processes) and unique files (files that were opened only by a subset of MPI processes). The I/O time for unique files ($\phi^U$) is calculated as the sum of the total time taken to do metadata operations ($\phi^U_M$) and the time taken to perform read and write operations ($\phi^U_{RW}$) by the slowest rank of all processes involved in I/O. The I/O time for the shared files ($\phi^S$) is calculated as the difference between the timestamps of the first file open of any shared file and the last I/O done on any shared file[1]. Darshan thus reports the total time spent by a job on I/O as

$$\phi = \phi^U + \phi^S = \phi^U_M + \phi^U_{RW} + \phi^S. \quad (1)$$

Storage-system-side information on Intrepid was obtained by using the iostat command-line tool included with the Sysstat collection of utilities [17]. The iostat tool reports the total system-wide read and write bandwidth as well as the number of read and write operations that are requested; this data is collected at the block device level on each storage server, with output reported over 60-second intervals. This data is summed across all storage servers for each interval and referred to as "system-wide I/O" in the subsequent sections because it represents an aggregate view of all I/O traffic on the storage system, regardless of how many applications were running. This iostat data was collected for Intrepid's two large parallel scratch file systems; system-side monitoring was not enabled for other shared file systems (including the home file system) that may have been accessed by a job on Intrepid. This data was collected over consecutive days from January 23 to March 26, 2010, with the exception of four days in February when the data was lost because of an administrative error.

The amount of system-wide activity happening during a particular job can be estimated by calculating the amount of system-wide I/O that occurred between the start and end of the job. We now formalize this concept. We let $\Delta$ denote the granularity (in seconds) of the system-wide I/O measurement (i.e., 60 seconds for the iostat data). We let $T_{\text{start}}$ and $T_{\text{end}}$ denote the start and end times (in seconds) of a job, respectively, and take $T_1, \ldots, T_{n-1}$ to be the iostat output timestamps between the job's start and end times:

$$T_0 \le T_{\text{start}} < T_1 = T_0 + \Delta < \ldots < T_{n-1} < T_{\text{end}} \le T_n.$$

[1]Subsequent versions of Darshan introduced a more direct measurement of I/O time for shared files, but it was not available when this data set was collected.

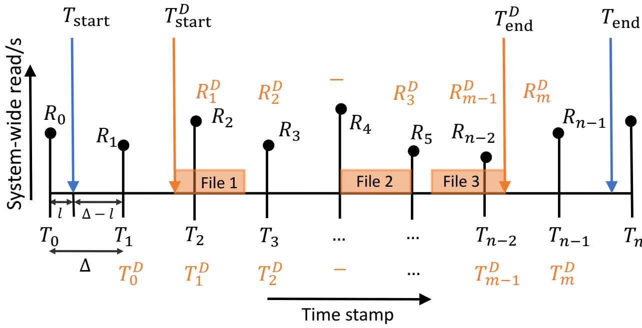


Fig. 1. Illustration of the calculation of the metrics $\psi_R$ and $\psi^D_R$.

For $i = 0, \ldots, n$, we let $R_i$ and $W_i$ denote, respectively, the system-wide read and write bandwidths reported at the end of interval $i$ (i.e., at time $T_i$). Similarly, we take $R^O_i$ and $W^O_i$ to denote, respectively, the number of read and write operations reported at time $T_i$.

Since we cannot obtain system-wide information at a granularity finer than $\Delta$, we perform linear interpolation and allocate the system-wide activity during the first and last iostat intervals according to the proportion of each interval for which the job was active; see Fig. 1 for an illustration. Hence, we estimate the total system-wide read and write volumes by

$$\begin{aligned}\psi_R &= \Delta \sum_{i=1}^{n} R_i - (T_{\text{start}} - T_0)\, R_1 - (T_n - T_{\text{end}})\, R_n \\ \psi_W &= \Delta \sum_{i=1}^{n} W_i - (T_{\text{start}} - T_0)\, W_1 - (T_n - T_{\text{end}})\, W_n.\end{aligned} \tag{2}$$

Similar expressions are used to estimate the number of system-wide operations:

$$\begin{aligned}\psi^O_R &= \sum_{i=1}^{n} R^O_i - \frac{T_{\text{start}} - T_0}{\Delta} R^O_1 - \frac{T_n - T_{\text{end}}}{\Delta} R^O_n \\ \psi^O_W &= \sum_{i=1}^{n} W^O_i - \frac{T_{\text{start}} - T_0}{\Delta} W^O_1 - \frac{T_n - T_{\text{end}}}{\Delta} W^O_n.\end{aligned} \tag{3}$$

Within a job, Darshan also captures information at an individual file level. For each file, the timestamps corresponding to the first and last read (write) operations, the amount of data read (written), and the corresponding file system mount point are each captured. This potentially finer-grained information allows one to consider only those iostat intervals in which a read/write occurs. Formally, we let $T^D_{\text{start}}$ and $T^D_{\text{end}}$ denote the times of the first and last I/O (i.e., read or write) operation for the job, respectively. By $T^D_1, \ldots, T^D_m$ we denote the (possibly nonconsecutive) iostat timestamps at the end of an iostat interval for which Darshan recorded some I/O activity for the job. $T^D_0$ denotes the timestamp of the end of interval immediately before $T^D_1$. We also let $R^D_i$ and $W^D_i$ denote, respectively, the system-wide read and write bandwidths reported at iostat timestamp $T^D_i$. A pair of volume estimates that uses this potentially smaller/nonconsecutive set

TABLE I
I/O CHARACTERISTICS FOR THE APPLICATION GROUPS CONSIDERED WITH I/O ON ALL FILES SYSTEMS (*I/O on file systems monitored by iostat*) (∗ DENOTES THAT THE SHARED FILE PERFORMS NO I/O).

| App. | Procs | Unique Files | Shared Files | Read (GiB) | Write (GiB) | Samples |
|---|---|---|---|---|---|---|
| ESB | 4096 | 57346 | 3 | 938.7 | 1.7 | 173 |
| | | *14338* | — | *234.8* | *0.5* | |
| NPC | 4096 | 10808 | 1 | 791.1 | 326.8 | 724 |
| | | *2744* | — | *195.3* | *81.7* | |
| NPB | 4096 | 3 | 1∗ | 4.3 | 1.1e-04 | 406 |
| | | *1* | — | *4.3* | — | |

TABLE II
MEAN (*standard deviation*) OF I/O TIME METRICS.

| App. | $\phi^U_M$ | $\phi^U_{RW}$ | $\phi^S$ | $\phi^D$ | $\phi^F_R$ | $\phi^F_W$ | $\phi^J$ |
|---|---|---|---|---|---|---|---|
| ESB | 1163.2 | 65.5 | 0.4 | 1171.9 | 137.8 | 1.2 | 1517.7 |
| | *214.1* | *20.0* | *0.1* | *191.4* | *13.1* | *3.8* | *287.5* |
| NPC | 149.5 | 274.4 | 28.5 | 3522.3 | 2628.4 | 248.1 | 3738.8 |
| | *132.4* | *41.1* | *4.8* | *302.1* | *244.8* | *54.0* | *334.3* |
| NPB | 0.4 | 23.8 | — | 1653.3 | 183.7 | 1290.8 | 2117.1 |
| | *0.6* | *63.7* | — | *74.8* | *8.9* | *63.1* | *63.6* |

of iostat intervals is then

$$\begin{aligned}\psi^D_R &= \Delta \sum_{i=1}^{m} R^D_i - \left(T^D_{\text{start}} - T^D_0\right) R^D_1 - \left(T^D_m - T^D_{\text{end}}\right) R^D_m \\ \psi^D_W &= \Delta \sum_{i=1}^{m} W^D_i - \left(T^D_{\text{start}} - T^D_0\right) W^D_1 - \left(T^D_m - T^D_{\text{end}}\right) W^D_m.\end{aligned} \tag{4}$$

A corresponding I/O time estimate accounts for the $m$ iostat intervals during which Darshan logged I/O activity:

$$\phi^D = m\Delta. \tag{5}$$

An analogous set of metrics is defined for the case where only the files stored on file systems monitored by iostat are considered. The respective measures for system-wide read and write volume are $\psi^I_R$ and $\psi^I_W$, and the I/O time metric is denoted by $\phi^I$.

The time spent performing reads (writes) for each file within a job can be estimated from the difference between the timestamps of the first and last read (write) reported by Darshan. For a file $f$ we denote these read and write times by $\phi^f_R$ and $\phi^f_W$, respectively. Another metric we explore is the maximum time over all files in the job:

$$\begin{aligned}\phi^F_R &= \max_f \phi^f_R \\ \phi^F_W &= \max_f \phi^f_W.\end{aligned} \tag{6}$$

## IV. APPLICATION GROUP PROPERTIES

The application groups used in this paper are from [2], in which the jobs monitored by Darshan are grouped based on the number of files, number of processors, amount of data read and written, and user ID. Since only 27% of the jobs running on Intrepid had Darshan logging enabled during the period under consideration, we focus on three application groups with ample Darshan data—Earth Science B (ESB), Nuclear Physics C (NPC), and Nuclear Physics B (NPB), respectively consisting

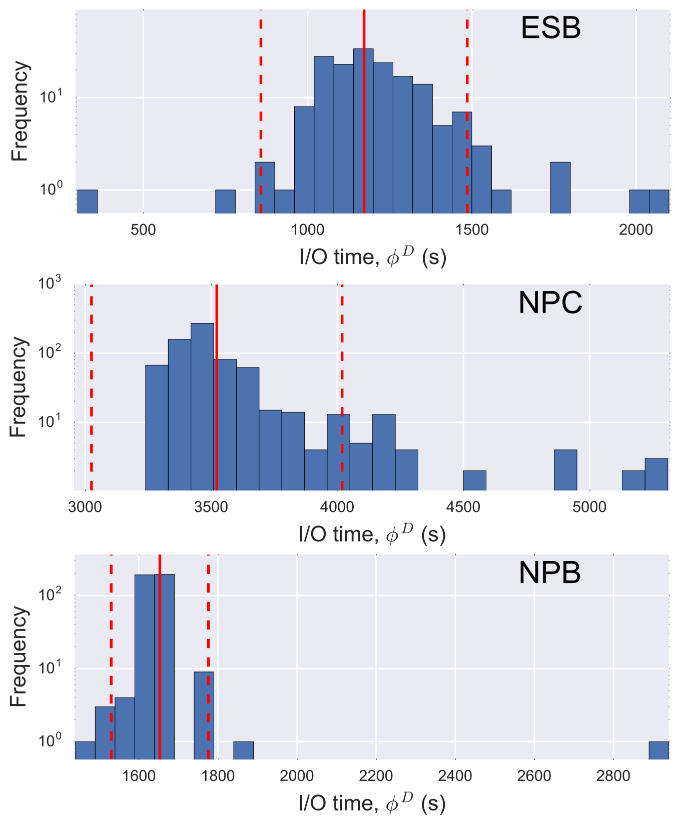


Fig. 2. Histograms of the I/O time $\phi^D$ for each application group; the solid line locates the mean and the dashed lines highlight 1.645 standard deviations on either side of the mean.

of 173, 724, and 406 jobs[2]—that have the same values for the features used in grouping; see Table I. All three groups are read heavy, with relatively small contributions from shared files. Although the jobs within an application group share the I/O feature values in Table I, the time the jobs use for I/O varies greatly within a group. Table II shows the means and standard deviations for the I/O metrics described in Sec. III, while Fig. 2 shows that the distributions of times (in this case, $\phi^D$) also exhibit significant skew. The solid red lines indicate the mean, and the dashed lines indicate mean $\pm 1.645$ standard deviations, which were the measures of variability used in [2].

The `ESB` jobs perform I/O on 57,346 unique files and have a high mean metadata time ($\bar{\phi}^U_M = 1163.2$ s) relative to the mean read/write time ($\bar{\phi}^U_{RW} = 65.5$ s) and mean run time ($\bar{\phi}^J = 1517.7$ s). A file-level view of the file activity during an example `ESB` job is shown in Fig. 3. We observe that the read operations occur in the first few seconds (about $1/15$ of the duration of the job's I/O time with the value of $\phi^F_R$ being 120 s), while the writes are concentrated toward the end of the job (with $\phi^F_W$ of 0.25 s). Since the file-level times tend to be small relative to the 60-second iostat intervals, we expect that the file-based system-wide volume measure $\psi^D$ will give a more accurate view of the system-wide I/O traffic that could

[2]The number of jobs differs slightly from the values reported in [2] because jobs were removed during periods when system-wide I/O volume information was missing (e.g., because the jobs ran outside the window in which data was collected or during the period when data was lost in February).

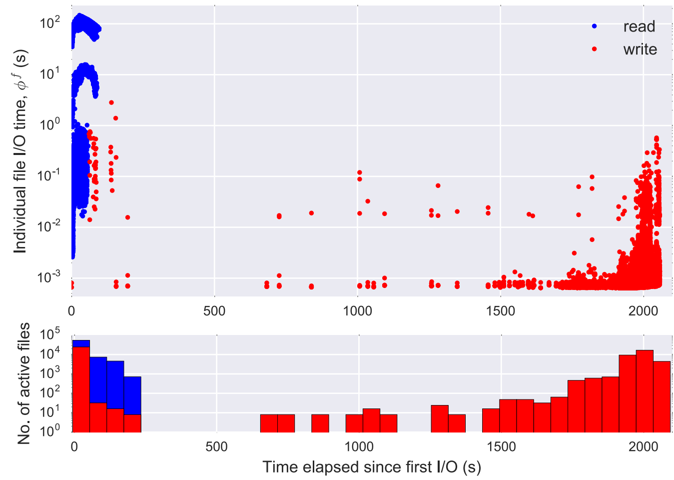


Fig. 3. File-level I/O activity from the start of an `ESB` job; there are 16,387 files with $\phi^f_R > 0$ and 32,770 files with $\phi^f_W > 0$.

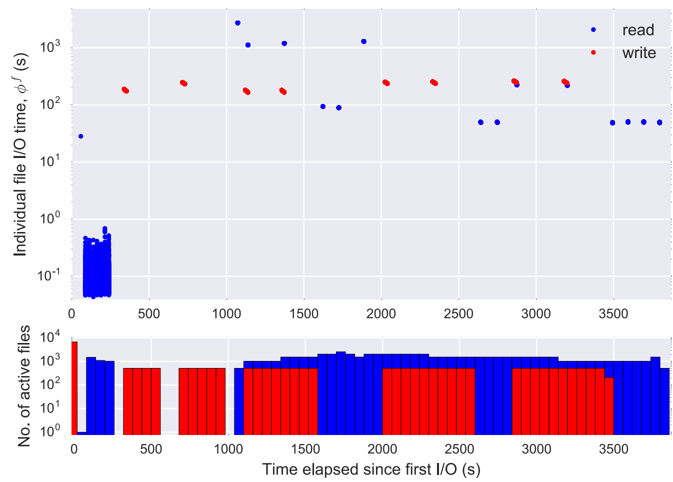


Fig. 4. File-level I/O activity from the start of an `NPC` job; there are 10,753 files with $\phi^f_R > 0$ and 4,096 files with $\phi^f_W > 0$.

interfere with this job than will the solely iostat-based measure $\psi$.

For `NPC`, the large number (10,808) of unique files associated with I/O results in a high value for the mean metadata time ($\bar{\phi}^U_M = 149.5$ s). However, the value of $\bar{\phi}^U_{RW}$ dominates $\bar{\phi}^U_M$ and $\bar{\phi}^S$ for this application group; thus, a job's slowest process spent more time on I/O operations involving unique files than it spent on metadata operations. The file-level view of an example `NPC` job in Fig. 4 shows that many files have large $\phi^f_R$ and $\phi^f_W$ values and cover many 60-second iostat intervals. In this case, the file-based metric $\psi^D$ is unlikely to have significantly more information than the iostat metric $\psi$ has. This situation is also present for `NPB`, where the only two files with significant I/O activity, one with $\phi^f_R = 186$ s and one with $\phi^f_W = 1288$ s (see Fig. 5), cover several 60-second iostat intervals.

We note that for `ESB` and `NPC` only 25% of the total files (approximately 25% of the total I/O volume) were accessed on the two file systems monitored by iostat. For `NPB`, however,

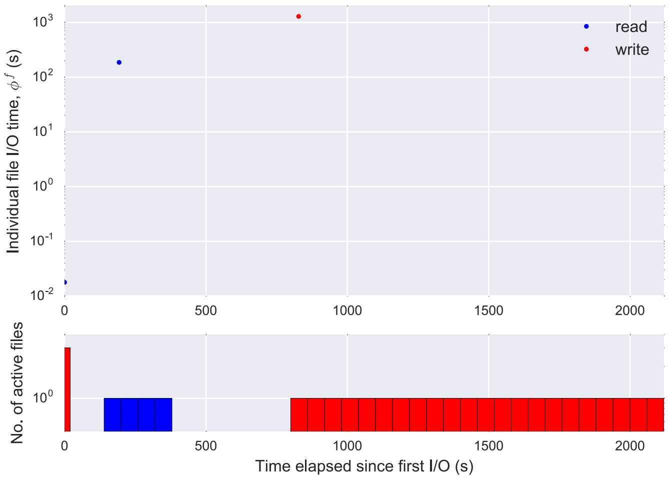


Fig. 5. File-level I/O activity from the start of an NPB job; there are 2 files with $\phi_R^f > 0$ and 1 file with $\phi_W^f > 0$.

only one file is involved in I/O on file systems monitored by iostat, and this file accounts for the bulk of the I/O volume.

## V. Methodology

We model the I/O time of a job $z$ on a given platform by

$$\phi_z = g(\eta_z, \nu_z, \beta_z), \tag{7}$$

where $\eta_z$ is the volume of I/O generated by the job, $\nu_z$ is the volume of system-wide I/O during job $z$ that does not include $\eta_z$, and $\beta_z$ represents factors that can impact $\phi_z$ but that remain unchanged, uncontrolled, and/or unobserved. These factors can include the effects of routing algorithms, network congestion, disk contention, and shared resources. Another factor can be the system-wide I/O on the file systems where monitoring was not present but can affect the application I/O performance through network resource sharing. Given an application group, $\eta_z$ is constant across all jobs. Therefore, the factors that cause the observed I/O variability are $\nu_z$, $\beta_z$, and our ability to measure each of the quantities. In what follows, we describe a systematic methodology that uses data

$$\{(\phi_{z_k}, \psi_{z_k}) : k = 1, \dots, d\} \tag{8}$$

from $d$ jobs of a selected application group to quantify connections between $\nu_z$ and $\beta_z$.

### A. I/O time variability and system-wide I/O traffic

As discussed in Sec. III, it can be difficult to obtain exact starting and ending points for I/O and thus difficult to separate $\eta_z$ and $\nu_z$. For this reason, we primarily use the model

$$\phi_z = g(\psi_z, \beta_z), \tag{9}$$

where $\psi_z = \eta_z + \nu_z$ denotes the total volume of system-wide I/O during the lifetime of the job; see Sec. III. Since $\eta_z$ is constant for a given application group, a relationship between $\psi_z$ and $\phi_z$ implies that $\phi_z$ is affected by $\nu_z$. Customary ways of measuring the relationship between two observed variables include correlation analysis (using both linear and nonlinear correlation methods) and model-based approaches.

*1) Linear correlation:* Linear correlation between two variables is typically measured by the Pearson product-moment correlation. Based on a line of best fit through the observed values, the Pearson correlation coefficient, $[-1 \le \rho \le 1]$, is a normalized measure of how far away the data points are from this line. Values of $\rho$ greater (less) than 0 indicate a positive (negative) association, and $|\rho| = 1$ indicates a perfect linear association. We note that values of $\rho$ close to zero do not necessarily indicate that the variables are independent since a nonlinear association may instead exist.

*2) Nonlinear correlation:* Nonlinearity can come in many forms, and no single method will be robust enough to detect all possible nonlinear relationships between two variables. Therefore, we consider several methods for identifying nonlinear association between two variables.

*Spearman's rank correlation coefficient ($r$)* determines the strength and direction of the monotonic relationship between two variables rather than the strength and direction of the linear relationship as given by Pearson's correlation coefficient. In a monotonic increasing (decreasing) relationship, one variable increases (decreases) as the other increases; such a relationship is nonlinear whenever the rate of increase (decrease) is not constant. The Spearman coefficient satisfies $-1 \le r \le 1$, and its interpretation of correlation strength is similar to the Pearson coefficient. Since it works with the ranks of the observations, however, the Spearman coefficient is less sensitive than the Pearson coefficient to outliers.

*Distance correlation ($\mathcal{R}$)* treats two variables as random variables and computes a statistical distance between the empirical probability distributions of the two variables. The distance correlation also satisfies $0 \le \mathcal{R} \le 1$; unlike the previous measures, $\mathcal{R} = 0$ if and only if $\phi_z$ and $\psi_z$ are independent.

*Mutual information score ($\mathcal{I}$)* measures the amount of information shared between $\phi_z$ and $\psi_z$ and how much the uncertainty about one variable is reduced by knowing the other variable [18]. We use a normed mutual information score that satisfies $0 \le \mathcal{I} \le 1$, where 1 indicates strong correlation and 0 indicates no correlation.

*3) Model-based approach:* The model-based approach (also called regression) is a class of supervised learning methods in which one uses an algorithm to estimate a functional form for the relationship between $\phi_z$ and $\psi_z$. One such algorithm used for regression is *random forest*, which is popular in the machine learning community [19]. It builds a number of decision trees, where each tree is obtained by using a random subset of the original dataset and splitting the input space recursively into a number of hyperrectangles. The obtained rectangles contain the input values that have similar outputs. The average of the outputs within a hyperrectangle is then assigned as the value for that hyperrectangle. Each tree follows an if-else rule and returns the constant value at the leaf as the predicted value for a new point $x^*$; the random forest output value at this point is the mean of all values predicted by the trees.

TABLE III
CORRELATION METRICS WITH I/O TIME ($\phi^D$).

| | ESB | | NPC | | NPB | |
|---|---|---|---|---|---|---|
| | $\psi_R^D$ | $\psi_W^D$ | $\psi_R^D$ | $\psi_W^D$ | $\psi_R^D$ | $\psi_W^D$ |
| Pearson | 0.55 | 0.42 | 0.19 | 0.23 | -0.05 | 0.04 |
| Spearman | 0.42 | 0.53 | 0.14 | 0.35 | -0.02 | -0.02 |
| Dist. corr. | 0.55 | 0.45 | 0.16 | 0.30 | 0.06 | 0.05 |
| Mutual info. | 0.64 | 0.51 | 0.30 | 0.54 | <1.0e-16 | <1.0e-16 |

### *B. I/O time variability and unknown factors*

The factors in $\beta_z$ typically cannot be measured or modeled directly; furthermore, our observations $(\psi_z, \phi_z)$ are subject to other sources of measurement error. Therefore, we model $\psi_z$ and $\phi_z$ as random variables; and we analyze the joint probability density function (PDF) $f(\phi_z, \psi_z)$, which is a statistical measure that captures the likelihood of the pair of events $(\phi_z, \psi_z)$ occurring together. One can also obtain conditional densities such as $f(\phi_z|\psi_z = \psi)$, which is the probability distribution of $\phi_z$ when $\psi_z$ is fixed to a particular value $\psi$. Conditional densities can be used for estimating quantities such as the mean, median, and uncertainty of $\phi_z$ as a function of $\psi_z$.

Kernel density estimation (KDE) is a method for approximating $f(\phi_z, \psi_z)$. It is essentially a data-smoothing approach in which the contribution of each observed data point is smoothed over a local neighborhood of that data point. The contribution of the data point $(\phi_{z_k}, \psi_{z_k})$ to the kernel density estimate at a generic point $(\phi, \psi)$ depends on the proximity of these two points. The extent of this contribution depends on the shape of the kernel function and an associated kernel bandwidth parameter. Here we use the standard Gaussian kernel function and the kernel bandwidth obtained by minimizing the asymptotic mean integrated squared error [20].

## VI. RESULTS AND DISCUSSIONS

We now apply the methods described in Sec. V to the ESB, NPC, and NPB application groups. We first examine the original data (obtained from a 60-second iostat measurement granularity) and then test the effect of changing this measurement granularity.

### *A. System-wide I/O at 60-second iostat measurement granularity*

*1) Correlation measures for association between I/O time ($\phi^D$) and system-wide I/O ($\psi^D$):* The results for ESB in Table III indicate moderately strong linear and monotonic correlations based on the values of Pearson's and Spearman's coefficients. The large value for the distance correlation metric indicates that the I/O time is not independent of the system-wide I/O. The higher value of mutual information as compared with the Pearson coefficient indicates a nonlinear correlation between the I/O time and system-wide I/O.

Similar analysis for NPC shows that the correlation between I/O time and the system-wide I/O is smaller than the corresponding values for ESB. As indicated by all the correlation metrics, $\phi^D$ has a higher correlation with $\psi_W^D$ than with $\psi_R^D$. The high value for the Spearman coefficient and mutual information compared with the Pearson coefficient indicates that the association between the quantities is highly nonlinear. The distance correlation values indicate that I/O time is not independent of the system-wide reads and writes.

The correlation between I/O time and the system-wide I/O for NPB is the smallest among the three application groups and close to zero for all the correlation metrics. The distance correlation values indicate that I/O time for this application group is close to being independent of the system-wide reads and writes.

These observations indicate that the I/O time is strongly correlated with the system-wide I/O (reads and writes) for the ESB application group with the nonlinear association. The correlation is also nonlinear but milder for NPC. For NPB, however, correlation is practically nonexistent and shows evidence of statistical independence between them.

*2) Correlation measures for association between I/O time ($\phi^I$) and system-wide I/O ($\psi^I$):* If we consider only the files located on monitored file systems (i.e., the two scratch file systems and not user home directories), then the correlations for ESB, NPC, and NPBfollow the same pattern observed in the preceding subsection but with a slight reduction in the strength of correlation. This result is somewhat counterintuitive, because we would not expect access activity on the monitored file systems to perturb performance for other resources. One possible explanation is that although the other file systems on Intrepid are provisioned to use separate storage servers and disk drives, they still share a common storage system network. High volumes of traffic on the unmonitored file systems can thus perturb I/O performance on the monitored file systems as a result of network resource sharing. This suggests that correlation strength could be improved beyond what we observe between $\phi^D$ and $\psi^D$ by incorporating system-wide I/O traffic on *all* the file systems that share network resources or by incorporating direct instrumentation of the shared network resources themselves.

*3) Correlation measures for association between metadata time ($\phi_M^U$) and number of system-wide I/O operations ($\phi^O$):* We also study the correlation between the metadata time $\phi_M^U$ and the system-wide I/O metrics based on the number of I/O operation counts ($\phi_R^O$ and $\psi_W^O$). The correlation metric values shown in Table IV consistently indicate a strong correlation for ESB. For this application group, however, $\phi_M^U$ has a higher correlation with $\psi_R^O$ than with $\phi_W^O$, which can be attributed to the fact that ESB reads from $16,387$ files in less than 240 seconds, thus generating a significant burst of metadata traffic. For NPC and NPB, $\phi_M^U$ shows a nonlinear correlation with $\psi_R^O$ and $\phi_W^O$ which is smaller than the corresponding values for ESB. This can be attributed to the fact that both the read and write activities involve file accesses that are more spread out temporally.

*4) Model-based method for association between I/O time ($\phi^D$) and system-wide I/O ($\psi^D$):* The random forest model is used to obtain the functional relationships $\phi^D = f(\psi_R^D)$ and $\phi^D = f(\psi_W^D)$ shown in Fig. 6. The trend of this functional relationship is seen to be nonlinear and monotonic,

TABLE IV
CORRELATION METRICS WITH METADATA TIME ($\phi_M^U$).

| | ESB | | NPC | | NPB | |
|---|---|---|---|---|---|---|
| | $\phi_R^O$ | $\psi_W^O$ | $\phi_R^O$ | $\psi_W^O$ | $\phi_R^O$ | $\psi_W^O$ |
| Pearson | 0.74 | 0.65 | 0.44 | 0.53 | 0.45 | 0.39 |
| Spearman | 0.80 | 0.75 | 0.58 | 0.67 | 0.61 | 0.57 |
| Dist. corr. | 0.79 | 0.74 | 0.52 | 0.62 | 0.57 | 0.50 |
| Mutual info. | 0.84 | 0.78 | 0.64 | 0.68 | 0.58 | 0.54 |

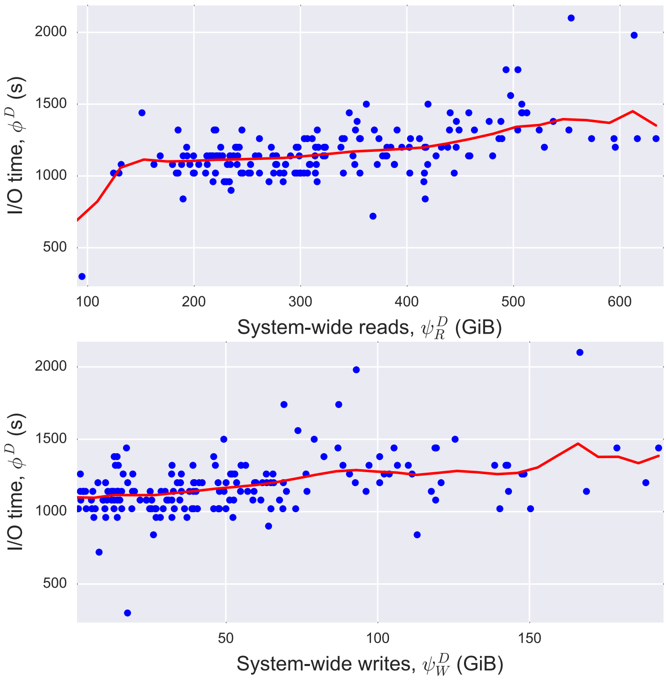


Fig. 6. Association between I/O time and system-wide I/O for ESB obtained using random forest.

which is consistent with the correlation metrics; however, it also provides additional information beyond the single value reported by the correlation metrics, such as the ability to predict the value of I/O time at a system-wide I/O value of interest.

Similar analysis for NPC and NPB showed similar results consistent with the correlation metrics, but the plots are not reported for brevity.

*5) KDE-based approach for association between I/O time ($\phi^D$) and system-wide I/O ($\psi^D$):* The PDF of the I/O time and the system-wide reads and writes for ESB obtained by using the KDE are shown in Fig. 7. The figure shows the contours of equal probability, where hotter colors indicate the regions with higher probability and the transition to cooler colors represents the decrease in probability. The solid white line indicates the median of $\phi^D$ as a function of $\psi_R^D$ (or $\psi_W^D$), and the white dotted line indicates the intervals that contain the central 90% of the probability mass (henceforth, the "credible intervals"). The solid red line indicates the mean, and the dashed lines indicate the mean $\pm 1.645\times$standard deviations (which can be interpreted as containing 90% of the probability mass for $\phi^D$ when ignoring its dependence on system-wide I/O).

This approach gives us information about the conditional

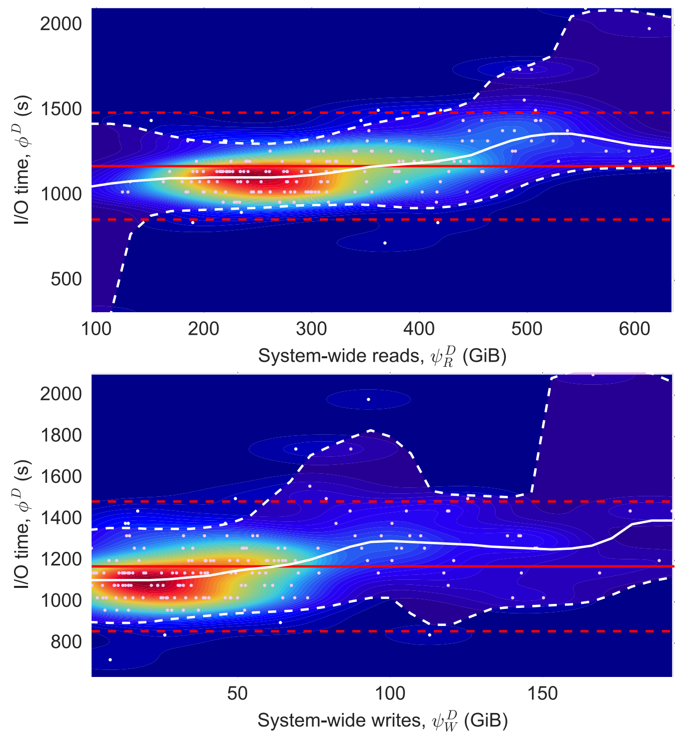


Fig. 7. KDE of the joint probability density for ESB shown as contours of equal probability (red=max, blue=min). The solid white line is the median, and the white dashed lines are the 90% credible intervals of $\phi^D$ as a function of $\psi_R^D$, $\psi_W^D$. The red solid (dashed) line represents the mean (mean$\pm 1.645\times$ the standard deviation) of $\phi^D$ ignoring its dependence on system-wide I/O.

probability distribution of the I/O time ($\phi^D$) at a particular system-wide I/O value (e.g., $f(\phi^D|\psi_R^D = 300$ GiB) for system-wide read volume), which can be used to obtain the median value and 90% credible intervals for $\phi^D$ when $\psi_R^D$ (or $\psi_W^D$) is fixed to a specific value (e.g., $\psi_R^D = 300$ GiB). Repeating this exercise for various values of $\psi_R^D$ gives the median response and 90% credible intervals of $\phi^D$ as a function of $\psi_R^D$.

The median response validates the moderately high positive correlation metrics obtained earlier, and this response is seen to be qualitatively similar to the random forest model response. We note that although the median of I/O time for ESB increases with the increase in the system-wide activity, the variability remains relatively constant, except in the regions of high system-wide activity where the data coverage is less. Also, the variability given by width of the 90% credible intervals is slightly smaller than the interval width (distance between the red dotted lines). This indicates that the I/O variability is slightly reduced when accounting for changes due to the system-wide activity.

The joint PDF models for NPC in Fig. 8 show that the median of $\phi^D$ with respect to $\psi_R^D$ and $\psi_W^D$ increases steadily; however, it is more prominent in the latter case. This observation is consistent with the correlation metrics that show the existence of nonlinear and higher correlation between $\phi^D$ and $\psi_W^D$.

Similarly, for NPB the median I/O time does not change with

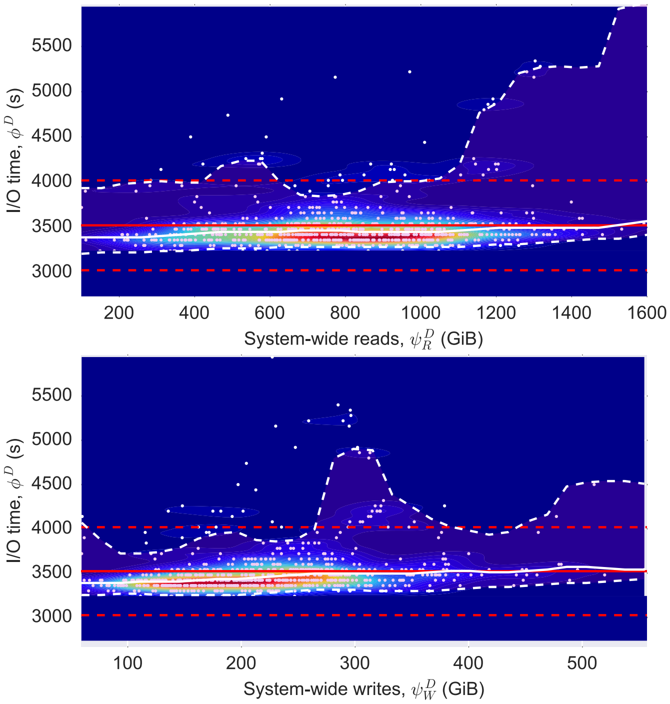


Fig. 8. Kernel density estimate plots for `NPC` (refer to caption of Fig. 7 for detailed description).

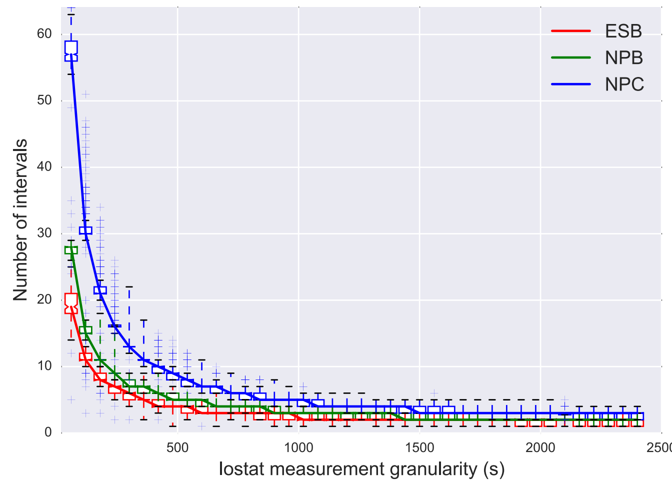


Fig. 9. Effect of iostat measurement granularity on number of iostat intervals used to calculate $\phi^D$.

the system-wide I/O, a result that agrees with the correlation measures that indicate minimal correlation between the I/O time and the system-wide I/O.

The change in the I/O time ($\phi^D$) with the system-wide I/O metrics ($\psi_R^D$ or $\psi_W^D$) behaves differently for the three application groups, as demonstrated by the results above. This behavior can be attributed to the inherent I/O characteristics of the job and the ability of the I/O time metrics to identify the time durations in which the job is performing I/O. Also, the file-server-side measurement granularity should be close to the smallest time interval in which contiguous I/O occurs. These criteria are well met for `ESB`, where the read I/O happens contiguously and in a short duration and, although the writes are more spread, they are contiguous and concentrated to the end. For this reason, we see a moderately high correlation for this application group. For `NPC`, there are fewer small, contiguous I/O and some large files where the location of exact I/O is not apparent. Thus, we see a decrease in the correlation measures compared with those of `ESB`. The `NPB` application group is an extreme case in which there are few but long files and the Darshan measure for time spent in doing read and write ($\phi_{RW}^U$) is significantly less than the duration of the individual files. Hence this application group shows smaller correlation than the other two groups do.

### *B. Effect of the iostat measurement granularity on the correlation metrics*

The system-wide I/O metrics ($\psi_z$) used in this study ($\psi_R^D, \psi_W^D, \psi_R^O, \psi_W^O$) depend on the ability to identify the exact intervals ($T_1^D, \ldots, T_{m-1}^D$) in which the I/O was performed by the application. The closer the actual I/O time inside the interval gets to the length of the interval, the more accurate the system-wide I/O gets. That is, the ratio of the I/O time in an interval to the length of the interval (granularity $\Delta$) ideally should be close to 1. Increasing the granularity to a higher value will reduce the the observed correlations, however high they might be. This behavior is demonstrated in this study by synthetically coarsening the iostat data by merging the information in multiple intervals, which is equivalent to taking measurements at a lower frequency.

Let the new granularity be $\Delta_k = k \times \Delta$, for $k = 1, 2, \ldots$. This choice changes the number of iostat intervals that fall inside the time duration of a job ($n$) and the number of iostat intervals involving I/O ($m$) and subsequently the metrics $\psi_R^D, \psi_W^D, \psi_R^N$, and $\psi_R^N$.

The granularity of the interval length $\Delta_k$ is increased from 60 to 1200 s, which corresponds to an increase in $k$ from 1 to 20. The change in the number of iostat intervals used to calculate $\phi^D$ with this increase in granularity for all three application groups is shown in Fig. 9.

Figure 10 show that although the correlation between the job I/O time metric ($\phi^D$) and the system-wide I/O metrics ($\psi_R^D$,$\psi_W^D$) is high for `ESB` at a 60-second granularity, this correlation decreases steadily with the increase in granularity. This behavior is true for both the linear and nonlinear correlation metrics considered in this study. Similar behavior is observed for the other groups, with lower values of the correlation coefficients (Fig. 11).

A similar study of the correlation between the metadata time for `ESB` ($\phi_M^U$) and the system-wide I/O metrics ($\psi_R^O$,$\psi_W^O$) shows a high correlation at the 60-second granularity, which also decreases steadily to a small correlation value (Fig. 12).

This behavior further signifies the importance of making the file-server-side measurements at a sufficiently fine granularity (relative to the I/O timescale for the particular job) so that the correlations in the job I/O time and the state of the file server can be identified reliably where they exist.

We note that those jobs for which Darshan logging was

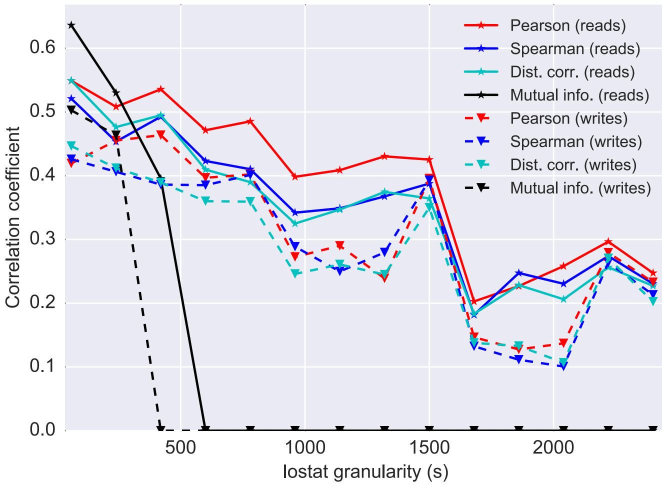


Fig. 10. Effect of iostat measurement granularity on correlation between $\phi^D$ and $\psi^D$ for `ESB`.

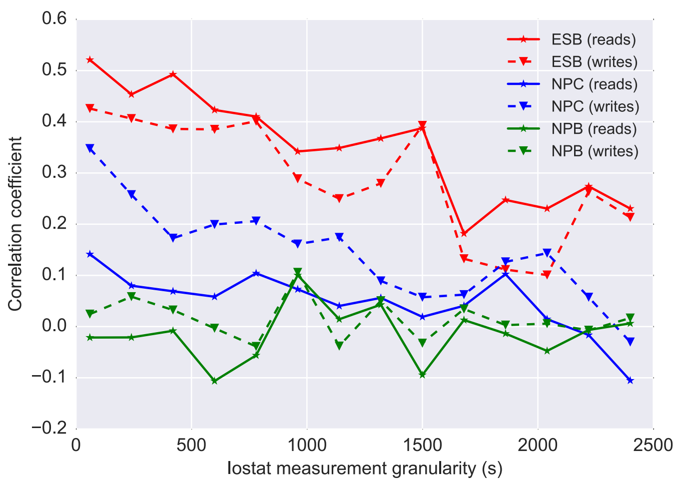


Fig. 11. Effect of iostat measurement granularity on Spearman correlation between $\phi^D$ and $\psi^D$ for all three application groups.

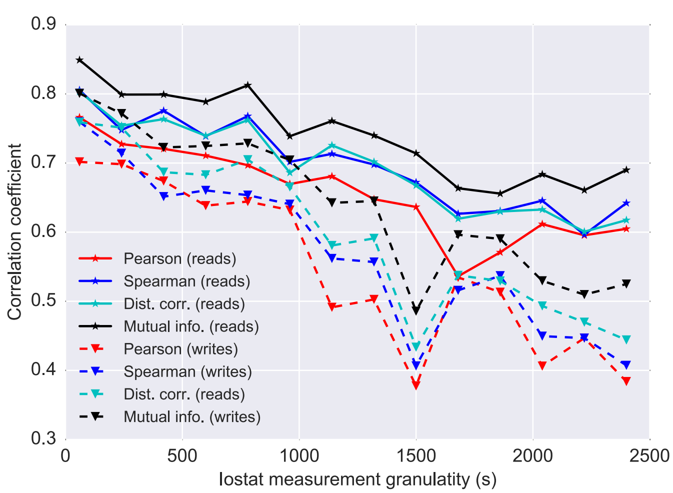


Fig. 12. Effect of iostat measurement granularity on correlation between $\phi^U_M$, $\psi^O_R$ for `ESB`.

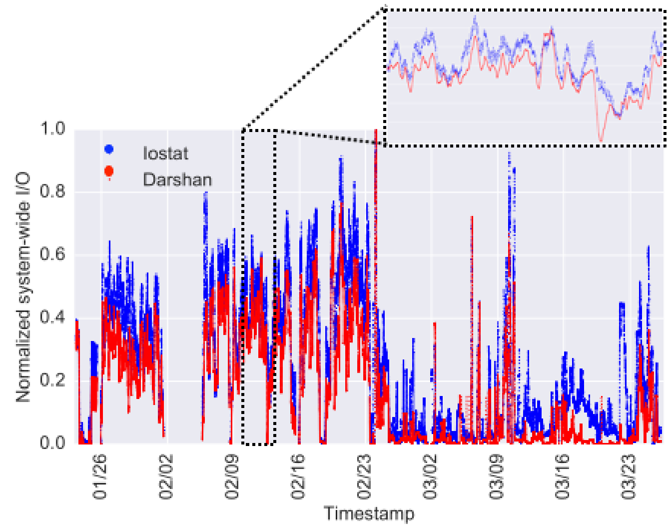


Fig. 13. Normalized system-wide I/O as measured by iostat (blue) and aggregate of available Darshan logs (red).

enabled accounted for 61% of the I/O volume recorded by iostat (7.8 PiB). Figure 13 shows the system-wide I/O volume measured by iostat and the aggregate I/O volume for the jobs logged by Darshan (with both volumes normalized by their respective maximum) and shows that the general trend is consistent among the two sources. With improved Darshan coverage, one would also have the prospect of using the system-wide Darshan logs to inform a system-wide I/O metric with a potentially finer temporal resolution. Such information could complement server-side information (e.g., from iostat).

## VII. Conclusions

We developed several I/O metrics and systematically applied statistical methods to analyze the relationship between a job's I/O time and the system-wide I/O traffic observed on the storage server. We applied our methodology to quantify the variability of the I/O time due to factors that cannot be measured or modeled directly, and we examined the relationship of this variability as a function of the system-wide I/O. We highlighted the importance of using appropriate measures (e.g., of the job I/O time and system-wide I/O) and correlation metrics when examining such relationships. We illustrated that the granularity in which I/O activities are measured is crucial for establishing a relationship between I/O time and system-wide I/O activity.

For the `ESB` application group, we found that I/O time is correlated with the system-wide reads and writes and that the I/O time variance remains relatively constant as the system-wide I/O changes. By accounting for the effect of system-wide I/O, the I/O variability due to unobserved factors is reduced relative to when this effect is not included. We found mild correlations for `NPC` and a behavior similar to `ESB`'s for the variability. In contrast, minimal correlation and near-probabilistic independence between the I/O time and the system-wide I/O was observed for `NPB` at the measurement granularity available. These findings confirm the existence of correlations between application I/O performance and system-wide I/O activity, but the nature of these correlations is not universal across all applications on the system. Real-world applications exhibit a diverse range of I/O strategies, and the relationship between application performance and storage system traffic differs according to each application's access pattern and I/O intensity.

Our results also highlight the importance of selecting appropriate metrics for I/O time. Application I/O time should be represented by discrete access time intervals for individual *files*, as shown in Eq. (5), rather than a single scalar value, in order to minimize noise from other phases of execution. System-wide metrics can be aligned with file I/O intervals by using linear interpolation to approximate data volumes in cases where the application monitoring granularity and storage system monitoring granularity do not match. Our evaluation of the impact of increased server-side sampling granularity suggests that the true correlations on today's I/O systems are likely stronger than those that can be calculated based on available measurements. To better understand and act on additional sources of I/O time variability, one would have to strike a balance between how much data to capture and how frequently, and the associated overhead of such measurement.

One of the challenges in this work was that the data presents an incomplete view of the application and server side. The ability to find higher correlations by using the method proposed in this work can be improved by improving server-side data measurement granularity; monitoring network resource utilization and sharing; monitoring server-side I/O on other file systems; and/or increasing the application coverage by Darshan. We plan to examine the degree to which finer-granularity I/O subsystem information could inform predictions for jobs with shorter I/O phases than considered here.

Although we focused on correlations, our approach can be easily extended to build models of I/O time that can be used to predict I/O time at system-wide I/O values of interest. Of particular interest is examining the effect of the number and diversity of samples on the predictive capability (that is, answering how many and under what types of system conditions, runs are needed in order to get a meaningful predictive model). This kind of analysis is readily extended to other file systems, such as Lustre, which require different (or more) server-side I/O metrics in order to analyze the interaction between system-wide I/O and application I/O time.

## Acknowledgment

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Contract DE-AC02-06CH11357.

This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.

## References

[1] D. Skinner and W. Kramer, "Understanding the causes of performance variability in HPC workloads," in *IEEE Workload Characterization Symposium*, Oct. 2005, pp. 137–149.

[2] P. Carns, K. Harms, W. Allcock, C. Bacon, S. Lang, R. Latham, and R. Ross, "Understanding and improving computational science storage access through continuous characterization," *Trans. Storage*, vol. 7, no. 3, pp. 8:1–8:26, Oct. 2011.

[3] O. Yildiz, M. Dorier, S. Ibrahim, R. Ross, and G. Antoniu, "On the root causes of cross-application I/O interference in HPC storage systems," in *2016 IEEE International Parallel and Distributed Processing Symposium*. IEEE, May 2016, pp. 750–759.

[4] A. Uselton, M. Hawison, N. Wright, D. Skinner, J. Shalf, L. Oliker, N. Keen, and K. Karavanic, "Parallel I/O performance: From events to ensembles," in *24th IEEE International Parallel and Distributed Processing Symposium*, 2010.

[5] Y. Liu, R. Gunasekaran, X. Ma, and S. S. Vazhkudai, "Server-side log data analytics for I/O workload characterization and coordination on large shared storage systems," in *International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov. 2016, pp. 819–829.

[6] S. E. Sayed, M. Bolten, D. Pleiter, and W. Frings, "Parallel I/O characterisation based on server-side performance counters," in *Joint International Workshop on Parallel Data Storage and Data Intensive Scalable Computing Systems (PDSW-DISCS)*, Nov. 2016, pp. 7–12.

[7] A. Traeger, E. Zadok, N. Joukov, and C. P. Wright, "A nine year study of file system and storage benchmarking," *Trans. Storage*, vol. 4, no. 2, pp. 5:1–5:56, May 2008.

[8] T. Hoefler and R. Belli, "Scientific benchmarking of parallel computing systems," in *International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov. 2015, pp. 73:1–73:12.

[9] B. Behzad, H. V. T. Luu, J. Huchette, S. Byna, Prabhat, R. Aydt, Q. Koziol, and M. Snir, "Taming parallel I/O complexity with auto-tuning," in *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, Nov. 2013, pp. 1–12.

[10] B. Behzad, S. Byna, S. M. Wild, Prabhat, and M. Snir, "Dynamic model-driven parallel I/O performance tuning," in *2015 IEEE International Conference on Cluster Computing (CLUSTER)*, Sep. 2015, pp. 184–193.

[11] F. Isaila, P. Balaprakash, S. M. Wild, D. Kimpe, R. Latham, R. Ross, and P. Hovland, "Collective I/O tuning using analytical and machine learning models," in *2015 IEEE International Conference on Cluster Computing (CLUSTER)*, Sep. 2015, pp. 128–137.

[12] J. Lofstead, F. Zheng, Q. Liu, S. Klasky, R. Oldfield, T. Kordenbrock, K. Schwan, and M. Wolf, "Managing variability in the IO performance of petascale storage systems," in *International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2010*. IEEE, 2010, pp. 1–12.

[13] M. Dorier, G. Antoniu, R. Ross, D. Kimpe, and S. Ibrahim, "CALCioM: Mitigating I/O interference in HPC systems through cross-application coordination," in *28th International Parallel and Distributed Processing Symposium*. IEEE, 2014, pp. 155–164.

[14] F. Isaila, J. Carretero, and R. B. Ross, "CLARISSE: A middleware for data-staging coordination and control on large-scale HPC platforms," in *IEEE/ACM 16th International Symposium on Cluster, Cloud and Grid Computing, (CCGrid 2016)*. IEEE Computer Society, May 2016, pp. 346–355.

[15] S. W. Son, S. Sehrish, W.-k. Liao, R. Oldfield, and A. Choudhary, "Reducing I/O variability using dynamic I/O path characterization in petascale storage systems," *Journal of Supercomputing*, pp. 1–29, 2016.

[16] P. Carns, "ALCF I/O data repository," Argonne National Laboratory, Technical Memorandum ANL/ALCF/TM-13/1, 2013.

[17] S. Godard, "Sysstat utilities home page," 2010. [Online]. Available: http://www.sebastien.godard.pagesperso-orange.fr/

[18] S. Khan, S. Bandyopadhyay, A. R. Ganguly, S. Saigal, D. J. Erickson III, V. Protopopescu, and G. Ostrouchov, "Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data," *Phys. Rev. E*, vol. 76, no. 2, p. 026209, 2007.

[19] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[20] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz, *Nonparametric and Semiparametric Models*. Springer Science & Business Media, 2012.