Exploring I/O Request Sizes of HPC Applications on a Supercomputer

Gessica Azevedo
Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
gessica.azevedo@inf.ufrgs.br

Jean Luca Bez
Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
jean.bez@inf.ufrgs.br

Pablo Pavan
Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
pablo.pavan@inf.ufrgs.br

Francieli Boito
LaBRI, Université de Bordeaux
Inria, CNRS, Bordeaux-INP
Bordeaux, France
francieli.zanon-boito@u-bordeaux.fr

Philippe Navaux
Institute of Informatics
Federal University of Rio Grande do Sul
Porto Alegre, Brazil
navaux@inf.ufrgs.br

***Abstract*—This study seeks to identify the most common input and output (I/O) request sizes used by HPC applications in supercomputers. We use data from the entire year of 2012 of characterization with the Darshan I/O profiling tool, in the Intrepid Blue Gene/P, to identify access patterns and request sizes. We contribute so that new optimization techniques can be evaluated considering these request sizes found in these environments. In general, we can conclude that of the six patterns chosen as a sample, five of them keep the trend of the average size of requests between 1 KB and 4 MB. Only one pattern stands out for its high variability for medium sizes, the MPI-IO write to a shared-file, due to optimizations provided by this interface.**

***Index Terms*—high performance computing, I/O, access pattern, request size, supercomputer**

I. Introduction

Scientific applications that run on high-performance computing systems (HPC) require high storage capacity and efficient data access. Parallel file systems (PFS) work by providing an abstraction of the storage system for these applications, which handle large amounts of data. In such a scenario, hundreds of computing nodes might need to access the shared storage system simultaneously. However, depending on how applications make their I/O requests, i.e., their access pattern, performance may be impaired. For instance, issuing small requests to the PFS might not offset the cost of accessing the remote storage system [1].

To apply optimization techniques to these applications, testing these techniques with benchmarks (such as MPI-IO Test [2], IOR[1], Ifer [3]) is essential but, we first need to understand the application's I/O behavior. Tools such as Darshan [4], developed at the Argonne Leadership Computing Facility (ALCF), provide characterization by creating application profiles. In order to identify the most common request sizes observed when issuing requests for the access patterns of applications on a supercomputer, this study analyzed Darshan data in Intrepid Blue Gene/P[2]. This paper's main contribution is to provide insights into what request sizes should be used when testing new optimization techniques, as those sizes will represent the sizes that HPC applications are issuing.

The remainder of this paper is organized as follows. Information on the data used in this study and its methodology are detailed in Section II. The results and analyzes are presented in Section III. Section IV discusses related work. Finally, in Section V, we conclude this paper and discuss future work.

II. Methodology

Between 2010 and 2013, ALCF collected execution data from various scientific applications using the I/O characterization tool called Darshan, in the Intrepid Blue Gene/P, ranked 23 in the November 2011 Top500 list. Darshan intercepts the flow of I/O functions and records a collection of statistics for each file that is opened [5]. The collected information allows identifying sizes and access patterns, operations, and time spent on I/O operations.

This information was used in a previous work [6], which extracted relevant data for this study. The focus of this study was on data generated in 2012 by the Darshan 2.0, resulting in $91,603$ jobs. The application coverage rate ranged from 20% to 80% per week [5], as Darshan only instrumented applications that successfully called `MPI_Init()` and `MPI_Finalize()`. However, this does not prevent applications from using POSIX, as both of these functions are used only to group information about the application's execution. We group the information by month and access patterns to identify the most common request size for each pattern and their behavior throughout the year. For this, we observe the minimum, median, maximum values, and quartiles.

Darshan continues to be used to collect information transparently, however, it is not common for these data to be public,

[1]https://github.com/hpc/ior

[2]https://www.top500.org/system/176322/

TABLE I
ACCESS PATTERNS REQUEST SIZES - SIX PATTERNS THAT MOST OCCURRED

	Access Patterns	Min. (Bytes)	Q1 (Bytes)	Median (KB)	Average (KB)	Q3 (KB)	Max. (MB)
A	POSIX, Write, Sequential Unique-file	1	99	15,7	3120,3	135	128
B	POSIX, Write, Consecutive Unique-file	1	4	0,003	0,019	0,003	16
C	POSIX, Read, Shared-file	1	160	0,27	93,6	64	256
D	POSIX, Read, Consecutive Unique-file	1	4096	4	12,6	4	16
E	POSIX, Write, Unique-file	1	2184	35,1	580,3	71,8	256
F	MPI-IO, Write, Shared-file	4	4194304	4096	6168,3	8192	122

for security and privacy reasons, as they involve information about users who performed the applications. It is not expensive to collect this profiling information, which makes several HPC centers transparently collect them in their supercomputers. Nonetheless, large updated data sets are often not public.

The 22 access patterns (described in Table II) observed throughout the year of 2012 can be classified into:

- Operation: write or read;
- File layout: single file or shared file;
- Spatiality:
  - Sequential: the `offset` does not have to be adjacent, but bigger than the previous one;
  - Consecutive: where the next `offset` to be accessed is immediately adjacent to the previous one;
- Interfaces: POSIX or MPI-IO.

Using the I/O phase estimates for the 22 patterns, we selected the six patterns observed during a longer period between applications, from these 22 patterns, as a sample to perform our analysis. These are detailed in Table I.

III. RESULTS AND ANALYSIS

In this section we present our results and analysis. Figure 1 illustrates the average request size for the six access patterns. The x-axis represents the days, and the y-axis represents the average size (in KB). We group the results by month and access patterns. The latter indicated by the letters of A-F. The description of these patterns is presented in the Table I.

For pattern A, it is observed that the average size remains between 1 KB and 3 MB, but some exceptions can reach up to 122 MB. Pattern B has a tendency for the average size up to 1 KB. However, a more significant variation between sizes was detected, mainly in the second half of the year. Some exceeded 1 MB, but most remained between 1 KB and 1 MB. In pattern C, there was also a concentration of sizes in the range of up to 1 KB, showing a variation increase in the last 3 months of the year. These are between 1 KB and 4 MB (except for a 7 MB). This behavior is also repeated for patterns D and E. The only difference is that for pattern E, the variations are around 24 MB, with some cases exceeding 73 MB and a maximum of 122 MB. The F pattern is the most distinctive, as it registered greater variability in sizes during the year.

In general, we can conclude that of these six patterns, five keep the average size of requests between 1 KB and 4 MB. The only pattern that stands out for its high variability for medium sizes is the pattern F, corresponding to the MPI-IO, Write Shared-file, and the reason for this flexibility of this pattern are the optimizations (like data sieving [7] and collective buffering [8]) that this interface provides.

TABLE II
ACCESS PATTERNS REQUEST SIZE - ALL PATTERNS

Access Pattern	Average (KB)	Median (KB)
POSIX, Write, Sequential, Unique-file	3120,3	15,7
POSIX, Write, Consecutive, Unique-file	0,019	0,004
POSIX, Read, Shared-file	93,6	0,27
POSIX, Read, Consecutive, Unique-file	12,6	4
POSIX, Write, Unique-file	580,3	35,1
MPI-IO, Write, Shared-file	6168,3	4096
POSIX, Read, Unique-file	27,9	0,5
POSIX, Write, Consecutive, Shared-file	1597,3	0,094
POSIX, Write, Shared-file	0,2	0,006
MPI-IO, Write, Independent, Shared-file	10,9	0,094
MPI-IO, Read, Collective, Shared-file	662,1	0,3
POSIX, Read, Consecutive, Shared-file	148,2	0,094
POSIX, Write, Sequential, Shared-file	3426,2	4096
MPI-IO, Read, Independent, Shared-file	22,1	0,5
MPI-IO, Write, Collective, Shared-file	1556,6	160,3
MPI-IO, Write, Unique-file	13393,6	4096
MPI-IO, Read, Collective, Unique-file	3119,5	511,1
MPI-IO, Write, Independent, Unique-file	0,083	0,004
POSIX, Read, Sequential, Unique-file	112,6	10
MPI-IO, Read, Independent, Unique-file	6734,7	978,5
MPI-IO, Write, Collective, Unique-file	12460,7	16384
POSIX, Read, Sequential, Shared-file	1407,8	1351,4

IV. RELATED WORK

Carns et al. analyzed the behavior of 66 applications on the Intrepid (ALCF) [9] supercomputer over two months of 2010. The authors demonstrated that the most recurring writing request size was between 100 KiB (kibibytes) and 1 MiB (mebibytes), while for read operations, it was between 100 bytes and 1 KiB. It was noticed that few applications influenced the observed access sizes. If these applications were disregarded in the analysis, the most frequent access size would change to 100 KiB and 1 MiB, for both operations.

Although previous work also used Intrepid's I/O load data, the study used a smaller data set. On the other hand, our research investigates the I/O behavior over an entire year (2012). It considers the size observed for the different access patterns and not only by the interface or operation.

Luu et al. [10] analyzed the Darshan logs of a million jobs representing a combined total of six years of I/O behavior across three leading HPC platforms. It was demonstrated that almost a third of the jobs had an aggregated throughput with a limit of 256 MB/s. Furthermore, three-quarters of the jobs only

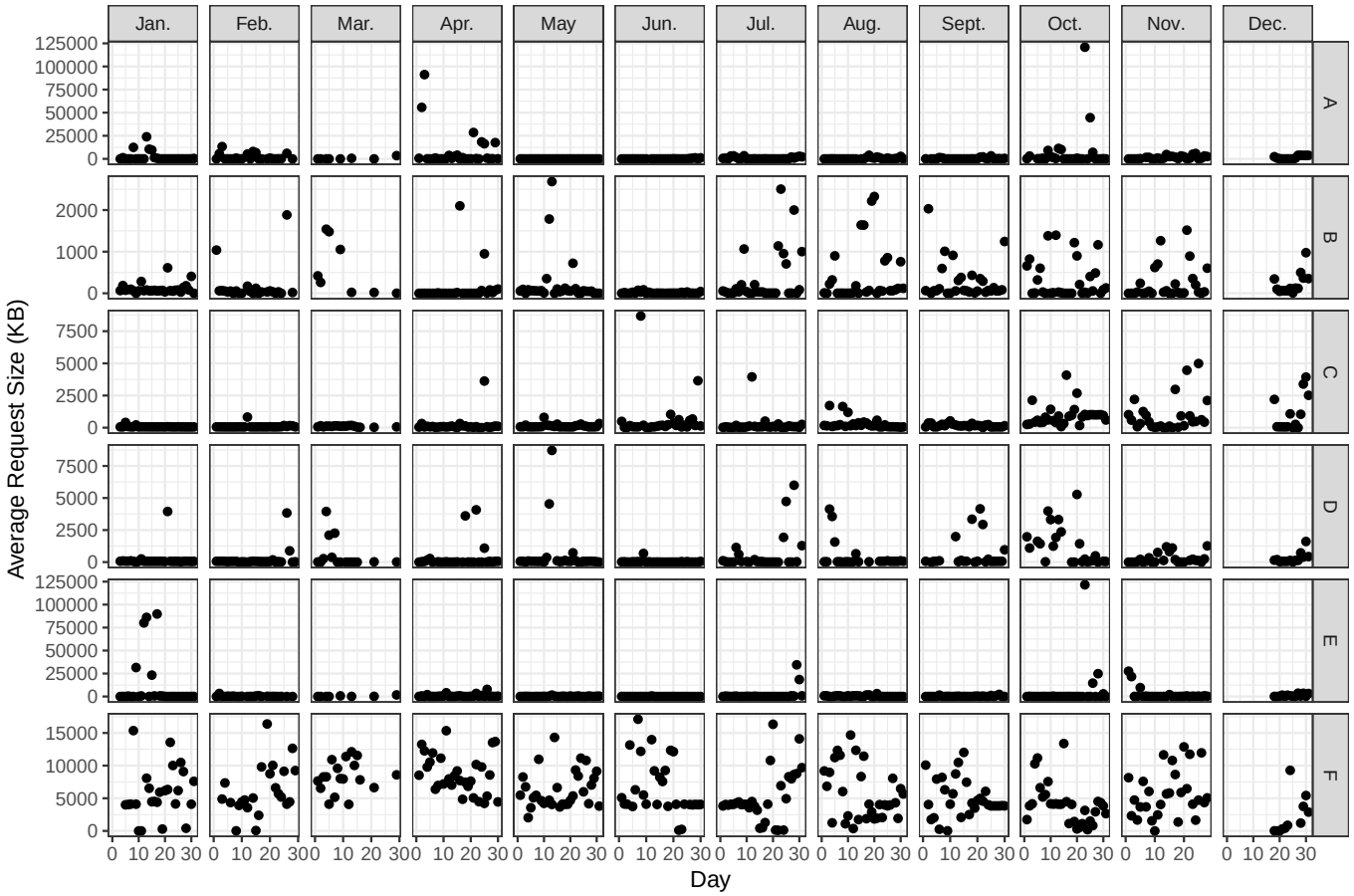

Fig. 1. Average request size (KB) over the days of a year. The y-axis is different for each pattern (row).

used POSIX to perform I/O, despite the existence of high-level parallel libraries.

Wang et al. [11] analyzed the IOR benchmark and two physics applications on a large Linux cluster, at the Lawrence Livermore National Laboratory (LLNL), with more than 800 dual-processor nodes. Each application presented only one or two typical request sizes. Large requests from several hundred KB to several MB were very common. However, small requests accounted for more than 90% of all requests, but the greatest amount of data was still transferred by large request sizes.

These two previous work also observed jobs from HPC applications, the first one focusing on the throughput of these jobs, while we focus on the request size, but in common, both could reach that most jobs used POSIX to perform I/O. The second one inspects the request sizes, same in our study, but we analyzed these sizes with the pattern associated. Besides, this work observed cases that small requests accounted for almost all requests. Our work also detected several instances with a great number of small requests.

V. Conclusion and Future Work

This study used data from an entire year of characterization with Darshan on the Intrepid Blue Gene/P supercomputer. It was possible to determine the most common I/O request sizes considering the different access patterns observed.

To evaluate new optimization techniques (initially using benchmarks such as MPI-IO Test, IOR, Ifer), it is necessary to use parameters close to reality so that the validation is consistent. In this way, identifying the most common request sizes helps the tests maintain reliability since these parameters will represent the reality of HPC applications. As future work, we intend to expand the analysis to the other patterns observed in the machine throughout the year.

Acknowledgment

The research received funding from PIBIC CNPq-UFRGS, from CAPES, grant N. 001, from CNPq and from the Petrobras project, grant N. 2016 / 00133-9. This research used resources from the *Argonne Leadership Computing Facility* at the Argonne National Laboratory, which is supported by the *Office of Science of the U.S. Department of Energy* under DE-AC02-06CH11357.

References

[1] F. Z. Boito, E. C. Inacio, J. Bez, P. O. A. Navaux, M. A. R. Dantas, and Y. Denneulin, "A Checkpoint of Research on Parallel I/O for High-Performance Computing." *In 2018 ACM Computing Surveys (ACM)*, Mar 2018.

[2] LANL, "Los alamos national lab mpi-io test, user's guide," 2006.
[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 *IPDPS 2016 - The 30th IEEE International Parallel and Distributed Processing Symposium*, Chicago, United States, May 2016. [Online]. Available: https://hal.inria.fr/hal-01270630
[4] P. Carns, R. Latham, R. Ross, S. L. K. Iskra, and K. Riley., "24/7 Characterization of petascale I/O workloads." *In 2009 IEEE International Conference on Cluster Computing and Workshops*, pp. 1–10, Aug 2009.
[5] P. Carns., "ALCF I/O Data Repository." Argonne Leadership Computing Facility, Tech. Rep., Feb 2013.
[6] P. J. Pavan, J. Bez, M. S. Serpa, F. Z. Boito, and P. O. A. Navaux, "An Unsupervised Learning Approach for I/O Behavior Characterization," *In 2019 31st International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, Oct 2019.
[7] R. Thakur, W. Gropp, and E. Lusk, "Optimizing noncontiguous accesses in mpi-io," *Parallel Computing*, vol. 28, no. 1, pp. 83–105, Jan. 2002. [Online]. Available: http://dx.doi.org/10.1016/S0167-8191(01)00129-6
[8] J. M. del Rosario, R. Bordawekar, and A. Choudhary, "Improved parallel i/o via a two-phase run-time access strategy," *ACM SIGARCH Computer Architecture News*, vol. 21, no. 5, pp. 31–38, Dec. 1993. [Online]. Available: http://doi.acm.org/10.1145/165660.165667
[9] 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." *In 2011 IEEE 27th Symposium on Mass Storage Systems and Technologies*, p. 7(3):8:1–8:26, May 2011.
[10] H. Luu, M. Winslett, W. Gropp, R. Ross, P. Carns, K. Harms, M. Prabhat, S. Byna, and Y. Yao, "A multiplatform study of I/O behavior on petascale supercomputers." *In Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing. ACM*, p. 33–44, 2015.
[11] F.Wang, Q. Xin, B. Hong, S. A. Brandt, E. L. Miller, D. D. E. Long, and T. T. Mclarty., "File system workload analysis for large scientific computing applications," April 2004.