A Study of Self-similarity in Parallel I/O Workloads

Qiang Zou*†, Yifeng Zhu‡, Dan Feng*
* *Huazhong University of Science and Technology, Wuhan, 430074 China*
† *University of Rhode Island, Kingston, RI 02881 USA*
‡ *University of Maine, Orono, ME 04469 USA*
E-mail: qzou@ele.uri.edu, zhu@eece.maine.edu, dfeng@hust.edu.cn

***Abstract*—A challenging issue in performance evaluation of parallel storage systems through trace-driven simulation is to accurately characterize and emulate I/O behaviors in real applications. The correlation study of inter-arrival times between I/O requests, with an emphasis on I/O-intensive scientific applications, shows the necessity to further study the self-similarity of parallel I/O arrivals. This paper analyzes several I/O traces collected in large-scale supercomputers and concludes that parallel I/Os exhibit statistically self-similar like behavior. Instead of Markov model, a new stochastic model is proposed and validated in this paper to accurately model parallel I/O burstiness. This model can be used to predicting I/O workloads in real systems and generate reliable synthetic I/O sequences in simulation studies.**

I. Introduction

Understanding I/O workload characteristic is critical in system modeling and simulation-based performance evaluation. Identifying representative I/O workloads allows us to fairly compare existing designs and faithfully evaluate new alternatives. Two basic approaches are widely taken to obtain representative workloads. One is to collect I/O traces in a production environment and then carefully reconstruct it during simulation [1]. The other is to use synthetic workloads that capture the behaviors of observed workloads and permit us to flexibly study the effects of some workload parameters. Both approaches require accurate statistical observations [2]. This paper aims to understand the parallel I/O characteristics of scientific applications that run in object storage systems.

An object data storage (ODS) is a new generation high performance parallel I/O architecture that promise unmediated host access to storage [3], [4]. Hence, data can flow in parallel between client hosts and object stores without passing any centralized server. Currently object storage systems, such as Luster [5] and Panasas [6], have been widely deployed in cluster supercomputers for scientific applications. This data access characteristics in object storages differ significantly from conventional network attached storages (NAS) that have been well studied. This is simply due to two important reasons: 1) Their workloads are substantially different. A NAS is typically designed for generic applications where parallel data accesses are rare while an ODS is usually for scientific applications that simultaneously run on a large number of computational nodes. 2) Data access mechanism changes I/O characteristics. While all data accesses are mediated by a centrally-managed file server in a NAS, an ODS can have many data paths flow independently without passing any single point of contact. Both factors lead to a larger degree of data parallelism and burstness in an OSD and this observation motivates us to analyze and model of its I/O burstness.

The traces analyzed in this paper, LLNL's scientific application traces, are collected in a Lustre cluster at the Lawrence Livermore National Laboratory (LLNL) [7] and mainly include three parallel scientific applications: *ior2*, *f1* and *m1*. Application *ior2* consists of three parallel I/O benchmarks, i.e., *ior2-fileproc*, *ior2-shared* and *ior2-stride*. LLNL's scientific applications simultaneously run on a large number of nodes in the Lustre [5] with more than 800 dual-processor nodes. Applications *f1* and *m1* are representative physics simulations. Both applications include two phases. While *f1* has *f1-restart* and *f1-write*, *m1* involves *m1-restart* and *m1-write*. These traces were collected in September 2003 and

978-1-4244-7153-9/10/ $26.00 @2010 IEEE

detailed description of these applications can be found in Ref. [8].

This paper analyzes and models the self-similarity of parallel I/O workloads for short-term scales. To the best of our knowledge, little research work has been made on this topic. Most of existing I/O burstiness studies are for general-purpose file systems [9]. This paper takes a first step to analyze the self-similarity in a set of scientific application I/O traces. The paper has the following conclusions.

1) The correlation study results show that it is necessary to further explore the self-similarity in parallel I/O workloads.
2) Parallel I/O workloads are self-similar for short-term time scales. Thus traditional Markov arrival processes are inappropriate to model the I/O demands.

Furthermore, we develop a stochastic model to accurately emulate or forecast future I/O arrival rates. We successfully validate our model by comparing the prediction results against the real traces. This model is helpful to generate synthetic I/O trace for performance evaluation and to predict future workloads for load balancing in real systems.

The rest of this paper is organized as follows. Section 2 gives an overview of the related works. Section 3 then explores several analytic tools to detect and analyze the self-similarity. We develop and validate a model to predict the I/O workload with self-similarity in Section 4. Section 5 concludes the paper.

II. Related Work

Prior research works have focused on the studies of synthesizing I/O workload both at the disk level [10], [11], [12], [13], [14], [15] and at the file system level [9]. At the disk level, the focus has been on trace synthesis [10], [11], [13], [14] and disk access pattern identification [11], [13], [14], [16], [17]. At the file system level, many studies provide useful insight into the design and analysis of various file systems for performance gains [8], [9]. In particular, Ref. [9] analyzes two sets of detailed, short-term application traces collected from general-purpose file systems, and finds that both exhibit self-similar like behaviors, with consistent Hurst parameters.

However, scientific applications tend to deviate significantly from commercial or generic applications in their I/O behaviors [18]. So far, several prior studies [8], [19], [20], [21] have analyzed the I/O behavior of parallel scientific applications, for tuning, managing, or optimizing parallel file systems. Ref. [8] examines the I/O burstiness of parallel I/O workloads using a simple methodology. They measure the *cumulative distribution functions* (CDF) of I/O inter-arrival times and conclude that I/O activities in the LLNL traces are very bursty in the *ior2* benchmark and the *f1* application. But F. Wang et al. hadn't further explored the characteristics in parallel I/O burstiness, such as the self-similarity.

Ref. [19] has proposed a Markov model to synthesize and predict I/O requests for scientific applications. Ref. [22] analyzes several I/O traces as well as used in Ref. [8], and concludes that correlations in parallel I/O inter-arrival times are inconsistent, either with little correlation or with evident and abundant correlations. Thus conventional Poisson or Markov arrival processes are inappropriate to model I/O arrivals in some applications. But Ref. [22] hadn't further studied the self-similarity in I/O-intensive parallel workloads.

This observation motivates us to re-examine the parallel I/O workloads studied in Ref. [8] and provide a rigorous statistical analysis to characterize the I/O burstiness in the following section.

III. Self-similarity Detection

In order to gain a deep understanding of workload behaviors, it is typically required to study the correlations of I/O inter-arrival times and characterize the I/O arrival patterns from a time dependence perspective first. In the following, we use *auto-correlation functions* (ACF) [23], [24] to study the patterns and characteristics in I/O inter-arrival times. The LLNL I/O traces studied in this paper are collected in many nodes. We find that the analytical results based on the traces collected on different nodes are very similar to one another in each scientific application. Therefore, this paper only presents the results of the traces collected at a randomly chosen node. The ACFs of these randomly selected traces are plotted in Fig. 1. As shown in Fig. 1, it might be appropriate to use an independently and identically distributed (IID) method such as Markov model to synthesize the *ior2-stride* workload, but not likely be useful in modeling the I/O requests in other workloads represented in Fig. 1. Examination results above

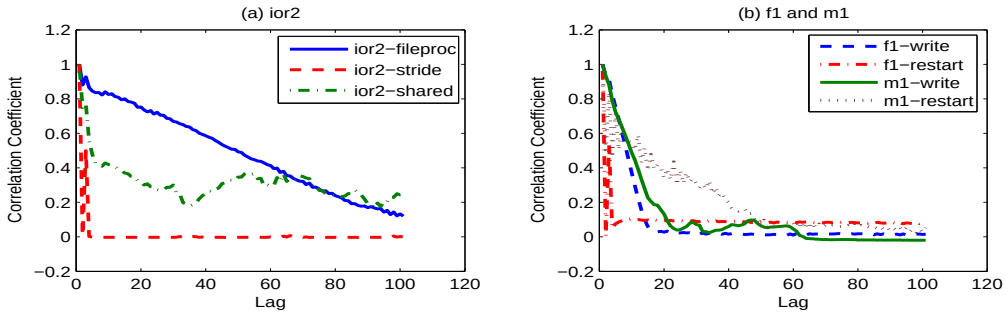

Fig. 1. From (a) to (b): ACFs of I/O inter-arrival times for *ior2*, *f1* and *m1*, respectively.

motivate us to further study the self-similarity of parallel I/O workloads in the following.

Degrees of self-similarity can be expressed as the speed of decay of series autocorrelation function using the *Hurst parameter*. Let $X = (X_t : t = 0, 1, 2, \ldots)$ be a covariance stationary stochastic process; that is, a process with constant mean $\mu = E[X_t]$, finite variance $\sigma^2 = E[(X_t - \mu)^2]$, and an $ACF(k) = E[(X_t - \mu)(X_{t+k} - \mu)]/E[(X_t - \mu)^2]$, $k \leq 0$, that depends only on k. We say the process X is self-similar if its ACF has the following property:

$$\lim_{k \to \infty} \frac{ACF(k)}{k^{-\beta}} = c < \infty, \quad \text{for} \quad 0 < \beta < 1. \qquad (1)$$

In particular, the process X is exactly second-order self-similar with Hurst parameter $H = 1 - \beta/2$ if the ACF is of the form:

$$ACF(k) = \frac{1}{2}[(k+1)^{2-\beta} - 2k^{2-\beta} + (k-1)^{2-\beta}]. \qquad (2)$$

The ACF of a self-similar process has an asymptotically hyperbolic decay as shown in Equation (1). Note that the ACF is non-summable, i.e. $\sum_k r(k) = \infty$.

The Hurst parameter, H, gives a measure of the degree of self-similarity of a given time-series. A value in the range $(0.5, 1)$ indicates the existence of self-similarity and a larger value implies a stronger temporal dependence. To estimate the Hurst parameter, we use one of the most popular analytic tools, R/S analysis (i.e., *Pox plot*), to detect and estimate the Hurst exponent to a set of observations. A brief description of the methods can be found in [13], [16], and a detailed analysis of various graphical and analytical methods are described in [25].

In the following, we use R/S analysis method to estimate the Hurst exponent values of LLNL I/O traces collected in many nodes. Without the loss of generality, we only present a few randomly selected streams in each trace in this paper. The Hurst exponent values of these randomly selected traces are shown in Table 1. All the estimates are the optimal estimates produced by a simple mathematical technique named least-square linear fitting. As shown in Table 1, most of the Hurst exponent values are above 0.5 except the I/O events in *ior2-stride*. This observation indicates the comprehensive existence of self-similarity in the LLNL I/O traces studied in this paper.

TABLE I
ESTIMATION OF H BY R/S ANALYSIS FOR I/O EVENTS ON *ior2*, *f1* AND *m1* TRACES, RESPECTIVELY.

Traces	Estimation of H				
Streams	1	2	3	4	5
ior2-fileproc	0.67	0.66	0.65	0.69	0.70
ior2-shared	0.72	0.78	0.83	0.84	0.86
ior2-stride	0.52	0.49	0.45	0.40	0.41
f1-write	0.55	0.56	0.54	0.58	0.56
f1-restart	0.51	0.59	0.52	0.50	0.64
m1-write	0.66	0.67	0.67	0.65	0.66
m1-restart	0.67	0.65	0.66	0.64	0.63

IV. SELF-SIMILARITY MODELING

A. Modeling Assumptions

In order to narrow down the huge number of potential models, we check the basic assumptions required by those models. We fortunately find that the parallel I/O traces studied meet the major requirements of a widely used model named Fractional Brownian Motion (FBM) [26], [27]. So we use FBM to synthetically model I/O workloads.

This paper defines a self-similar stochastic process by $A(t) = \lambda t + \sqrt{\alpha\lambda} Z(t)$, where $A(t)$ denotes the amount of I/O requests that has arrived at the storage pool in the time interval $[0, t)$, λ is the I/O mean arrival rate, α is a parameter related

to the variance of $A(t)$, and $Z(t)$ is the FBM process, characterized by the Hurst parameter H, t is a random variable with normal distribution, zero mean, variance$=1$. Then, $A(t)$ is a random variable, with normal distribution and zero mean. In particular, we always have $A(0)=0$. Let $A(t)$ be defined for all $t \in (-\infty, \infty)$ and denote the amount of I/O requests offered in the time interval $[s,t)$ by $A(s,t)=A(t)-A(s)$.

Norros' resulting formula for effective bandwidth can be used to estimate the bandwidth requirement for a self-similar traffic. The Norros effective bandwidth formula is more promising than previous effective bandwidth formulas because the degree of self-similarity is a parameter in the formula. The formula is briefly describe as (5):

$$C = \lambda + [-2\alpha\lambda \ln(\epsilon)((1-H)^{1-H}H^H)^2 \cdot b^{2H-2}]^{\frac{1}{2H}}, \quad (3)$$

where λ is I/O mean arrival rate, H is the Hurst parameter, α is the variance coefficient, C is the I/O effective bandwidth, ϵ is the I/O miss probability, and b is the maximum number of outstanding streams served per device.

The I/O miss probability ϵ can be expressed as following:

$$\lambda \sim \frac{C\epsilon}{b(1-\epsilon)}. \quad (4)$$

The prediction algorithm can be obtained after Equation (3) is substituted into Equation (4):

$$\lambda^{2H-1} \sim \frac{-2\alpha \ln(\epsilon)((1-H)^{1-H}H^H)^2 b^{2H-2}}{[b(1-\epsilon)-\epsilon]^{2H}}. \quad (5)$$

B. Experiments and Evaluations

Our prediction model can be briefly described as below.

ARRIVAL-RATE-PREDICTION

INPUT: The I/O miss probability ϵ, the maximum number of outstanding streams served at per node b, original trace data file f.
OUTPUT: arrival rate $\{\lambda(i); i=1,2,\ldots,n\}$.
ALGORITHM:
for each f
  Use maximum-likelihood estimate to estimate the parameter value α for data sets in f;
  Use Pox plot to estimate the Hurst value H
  if $H \notin (0,1)$ or $H = 1/\alpha$
    then break;
  else
    Set the initial values of ϵ and b, and obtain $\{\lambda(i); i=1,2,\ldots,n\}$ using Equation (5)
end for

In this section, we compare the accuracy of our prediction model against the realistic LLNL I/O traces collected in many nodes. We find that the analytical results based on the traces collected on different nodes are very similar to one another in each scientific application. Therefore, we only present the comparison results of one randomly selected stream in *f1-restart*.

Figure 2 compares the predicted arrival rates with the rates measured in the real trace with a sampling period of 0.01. The solid line represents the real trace, whereas the dashed is the synthetic trace generated by our prediction model. As we expect, the synthetic trace can accurately capture most burstiness in the real trace. A useful statistic tool named Fisher's analysis of variance (ANOVA) [28] reports the total variance is only 0.138, indicating our predicted arrival sequence can accurately emulate the actual arrival sequence. Furthermore, the one-way ANOVA analysis shows that both sequences have exactly the same mean arrival rate.

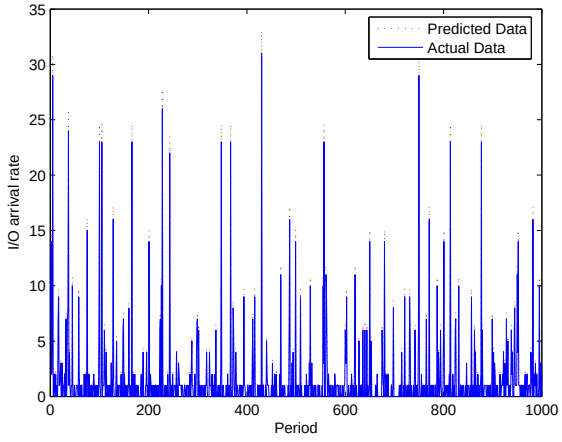

Fig. 2. Comparisons between actual arrival rates with predict model rates.

Our prediction model is very useful practically. One example is that our model can be used to generate synthetic I/O traces for simulation-based performance evaluation. Another example is that our model can be potentially used in real storage systems for resource allocation and load balancing. In data-intensive applications, the performance bottleneck is most often caused by load unbalance among all storage devices. If the future I/O arrival rate of each I/O stream can be precisely forcasted, then the aggregate I/O bandwidth of all available storage devices can be dynamically allocated among all streams to minimize the average response time. This technique is particularly useful in distributed parallel storage system where multiple data stores concurrently serving large amount of I/O requests.

V. Conclusions

One fundamental step in find solutions to alleviating the I/O performance bottleneck in high performance computing systems is to accurately characterize the I/O demands of scientific application workload. This paper analyzes a set of real I/O traces of scientific applications running in large supercomputers with object-based data storage systems. Our study has the following conclusions.

1) The self-similarity study is necessary for I/O-intensive scientific applications due to the fact there are evident correlations between inter-arrival times in subtraces collected on most computing nodes.
2) Similar to convention file system workloads, scientific I/O workloads are also self-similar for short-term scales. Thus traditional Poisson or Markovian arrival processes are inappropriate to model the I/O demands.

Additionally, we develop an accurate analytical model to predict I/O mean arrival rate for I/O workloads with self-similarity. We successfully validate our model by comparing the prediction results against the real traces. This model is useful to generate synthetic I/O trace for performance evaluation and also to predict future workloads for load balancing in real systems.

One limitation of this study is that all traces studied last from tens of seconds to half an hour. These short-period traces do not allow us to examine the self-similarity for long-term time scales. Our immediate future work is to collect parallel I/O traces lasting weeks or months and further evaluate longer-term self-similarity.

Acknowledgment

We would like to thank the anonymous reviewers for their helpful comments in reviewing this paper. We also thank LLNL for providing accesses to various tools and traces. Thanks go also to Zhihua Zhou, Guo Jiang, Dong Li and Zhidong Wang for their help in writing this paper. This research is supported in part by the National High Technology Research and Development Program (863Program) of China under Grant No.2009AA01A402, Changjiang innovative group of Education of China under Grants No. IRT0725, NSFC No.60933002, and the US NSF under Grants IIS-0916663, CCF-0937988, CCF-0621493, CCF-0811333 and CPS-0931820.

References

[1] L. Tian, D. Feng, H. Jiang, and et al., "Pro: A popularity-based multi-threaded reconstruction optimization for raid-structured storage systems," in *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST)*, San Jose, California, 2007.

[2] B. Anderson, "Mass storage system performance prediction using a trace-driven simulator," in *Proceedings of the 13th NASA Goddard/22nd IEEE Conference on Mass Storage Systems and Technologies (MSST'05)*, Monterey, California, April 2005.

[3] F. Wang, S. Brandt, E. Miller, and et al., "Obfs: A file system for object-based storage devices," in *Proceedings of the 12th NASA Goddard/21st IEEE Conference on Mass Storage Systems and Technologies (MSST'04)*, Greenbelt, Maryland, April 2004.

[4] W. K. For and W. Y. Xi, "Adaptive extents-based file system for object-based storage devices," in *Proceedings of the 14th NASA Goddard/23rd IEEE Conference on Mass Storage Systems and Technologies (MSST'06)*, College Park, Maryland, May 2006.

[5] "http://www.lustre.org/," *www.lustre.org/docs/whitepaper.pdf*.

[6] "Object storage architecture: Defining a new generation of storage systems built on distributed, intelligent storage devices," Panasas White Paper, Tech. Rep., 2003.

[7] "Lawrence livermore national labs," *http://www.llnl.gov/*.

[8] F. Wang, Q. Xin, B. Hong, and et al., "File system workload analysis for large scale scientic computing applications," in *Proceedings of the 12th NASA Goddard/21st IEEE Conference on Mass Storage Systems and Technologies (MSST'04)*, Greenbelt, Maryland, April 2004.

[9] S. Gribble, G. Manku, and E. Brewer, "Self-similarity in high-level file systems: Measurement and applications," in *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/Performance'98)*, Madison, Wisconsin, June 1998.

[10] Z. Kurmas, K. Keeton, and K. Mackenzie, "Synthesizing representative i/o workloads using iterative distillation," in *Proceedings of the 11th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'03)*, Orlando, Florida, October 2003.

[11] B. Hong and T. Madhyastha, "The relevance of long-range dependence in disk traffic and implications for trace synthesis," in *Proceedings of the IEEE Conference on Mass Storage Systems and Technologies (MSST'05)*, Monterey, CA, 2005.

[12] M. Gomez and V. Santonja, "Self-similarity in i/o workload: Analysis and modeling," in *Proceedings of the IEEE International Workshop on Workload Characterization (WWC'98)*, Dallas, Texas, 1998.

[13] ——, "Analysis of self-similarity in i/o workload using structural modeling," in *Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, College Park, Maryland, 1999.

[14] ——, "A new approach in the modeling and generation of synthetic disk workload," in *Proceedings of the 9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, San Francisco, California, 2000.

[15] A. Riska and E. Riedels, "Long-range dependence at the disk drive level," in *Proceedings of the Third International Conference on the Quantitative Evaluation of Systems (QEST'06)*, University of California, Riverside, California, 2006.

[16] M. Wang, T. Madhyastha, and et al., "Data mining meets performance evaluation: Fast algorithms for modeling bursty traffic," in *Proceedings of the 16th International Conference on Data Engineering (ICDE)*, San Jose, California, 2002.

[17] M. Wang, A. Ailamaki, and C. Faloutsos, "Capturing the spatio-temporal behavior of real traffic data," in *Proceedings of the IFIP International Symposium on Computer Performance Modeling, Measurement, and Evaluation*, Italy, 2002.

[18] S. Alam and J. Vetter, "An analysis of system balance requirements for scientific applications," in *Proceedings of the International Conference on Parallel Processing (ICPP'06)*, Columbus, Ohio, 2006.

[19] J. Oly and D. Reed, "Markov model prediction of i/o request for scientific application," in *Proceedings of the 16th Annual ACM International Conference on Supercomputing (ICS'02)*, New York City, New York, June 2002.

[20] E. Smirni and D. A. Reed, "Workload characterization of input/output intensive parallel applications," in *Proceedings of the 9th International Conference on Modeling Techniques and Tools for Computer Performance Evaluation*, St. Malo, France, June 1997.

[21] N. Tran, "Automatic arima time series modeling and forecasting for adaptive input/output prefetching," Ph.D. dissertation, University of Illinois at Urbana-Champaign, USA.

[22] D. Feng, Q. Zou, H. Jiang, and et al., "A novel model for synthesizing parallel i/o workloads in scientific applications," in *Proceedings of the IEEE International Conference on Cluster Computing (Cluster'08)*, Tsukuba, Japan, September 2008.

[23] J. Zhang, A. Sivasubramaniam, H. Franke, N. Gautam, Y. Zhang, and S. Nagar, "Synthesizing representative i/o workloads for tpc-h," in *Proceedings of the Tenth International Symposium on High Performance Computer Architecture (HPCA-10)*, Madrid, Spain, February 2004.

[24] A. Dainotti, A. Pescape, and G. Ventre, "Worm traffic analysis and characterization," in *Proceedings of the IEEE International Conference on Communications (ICC'07)*, Glasgow, Scotland, June 2007.

[25] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic (extended version," *IEEE/ACM Transactions on Networking*, vol. 2, pp. 1–15, Feb. 1994.

[26] Norros, "On the use of fractional brownian motion in the theory of connectionless networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 15, pp. 200–208, 1997.

[27] C. Stathis and B. Maglaris, "Modelling the self-similar behaviour of network traffic," *Computer Networks*, vol. 34, pp. 37–47, 2000.

[28] Z. J. Liu and et al., *Computational Science Technique and Matlab*. Beijing, P. R. China: Science Press, 2001.