2010 Ninth International Conference on Grid and Cloud Computing

Performance Evaluation of Parallel File System Based on Lustre and Grey Theory

Tiezhu Zhao
Guangdong Key Laboratory of Computer Network,
South China University of Technology,
Guangzhou, 510641, P.R.China
zhao.tiezhu@mail.scut.edu.cn

Jinlong Hu*
Guangdong Key Laboratory of Computer Network,
South China University of Technology,
Guangzhou, 510641, P.R.China
jlhu@scut.edu.cn

***Abstract*—High Performance Computing (HPC) system need to be coupled with efficient parallel file systems, such as Lustre file system, that can deliver commensurate IO throughput to scientific applications. It is important to gain insights into the deliverable parallel file system IO efficiency. In order to gain a good understanding on what and how to impact the performance of parallel file systems. This paper presents a study on performance evaluation of parallel file systems using Lustre file system. We conduct an in-depth survey on the basic performance factors of Lustre. Based on this survey, a series of test cases are designed to validate the performance of Lustre and we adopt relational analysis model and grey prediction model to analyze and predict the performance changes. In our relational analysis, we find that the performance of Lustre has a more closed correlation when performance factors change. Our prediction results indicate that our prediction model can obtain better prediction precision and could be further applied to performance evaluation of other parallel file systems.**

***Keywords*—Lustre, parallel file system, performance evaluation, performance model**

I. INTRODUCTION

Parallel file system is widely used in clusters dedicating to I/O-intensive parallel applications. The rapid development of HPC application is aggressively pushing the demand of parallel/distributed file system with high aggregated I/O bandwidth, mass storage capacity and high data fault-tolerant etc. However, the construction of parallel file system is much more expensive and complex. When a parallel file system is not properly tuned or configured, this cost may not be paid off. So, the issues on how to optimize the design of a parallel file system, how to evaluate the performance of a parallel file system, how to tune the performance of a parallel file system and how to predict the performance are more and more concerned by both storage industry and research communities.

Lustre is an open-source parallel file system. It is best known for powering seven of the ten largest HPC clusters in the world [1]. Lustre is a POSIX-compliant, object-based parallel file system. It is provided fine-grained, scalable I/O accesses to many storage targets. Lustre separates essential file system activities into three components: clients, metadata servers MDSes) that server metadata targets (MDTs) and manage Lustre metadata, and object storage servers (OSSes) that serve client's object request and object storage targets (OSTs) that serve backend storage as objects, respectively.

Although the various performance characteristics in HPC workloads have been studied via experimental analysis and empirical analysis, the potential performance of such systems can be difficult to predict because the potential impact to application performance in parallel file system environment is not clearly understood and most internal details of the basic components of parallel file system are not public.

Based on the above observations, we propose an in-depth performance evaluation based on Lustre file system and our evaluation mainly covers many performance factors, such as the number of OSSes, storage connection approaches, the type of disks, the type of journal for OST and the number of threads/OST etc. In order to obtain a good understanding on what and how to impact the performance of Lustre file system, we present grey relational analysis to mining the characteristics of the performance and deliver the grey performance prediction model to reveal the implications of parallel file system.

The rest of the paper is organized as follows. We firstly introduce related work in section II. In section III, we conduct an in-depth survey on performance factors of Lustre file system. Then, based on this survey, we present relational analysis model and grey prediction model. In section IV, we carry out a detailed performance analysis by our models and conclude the paper in section V.

II. RELATED WORK

Currently, these works in parallel/distributed file system can be divided into six categories: (1) Metadata management and query optimization. Metadata management is critical in scaling the overall performance of large-scale data storage systems and a large-scale distributed file system must provide a fast and scalable metadata lookup service. E.g. Wang et. al. proposed a two-level metadata management method to achieve higher availability of the parallel file system while maintaining good performance [2]; (2) Performance parameter Analysis and tuning. Yu et. al. indicated excessively wide striping can cause performance. To mitigate striping overhead and benefit collective IO, authors proposed two techniques: split writing and hierarchical striping to gain better IO performance [3]. Yu et. al. presented an extensive characterization, tuning, and optimization of parallel I/O on the Cray XT supercomputer

978-0-7695-4313-0/10 $26.00 © 2010 IEEE
DOI 10.1109/GCC.2010.34

(named Jaguar), and characterized the performance and scalability for different levels of storage hierarchy [4]; (3) Optimizing data distribution strategy. e.g. Li et. al. modeled the whole storage system's architecture based on closed Fork-Join queue model and proposed an approximate parameters analysis method to build performance model [5]. Yu et. al. adopted a user-level perspective to empirically reveal the implications of storage organization to parallel programs running on Jaguar and discovered that the file distribution pattern can impact the aggregated I/O bandwidth [6]; (4) Optimizing data access path. Piernas et. al. adopted a novel user-space implementation of Active Storage for Lustre and the user-space approach has proved to be faster, more flexible, portable, and readily deployable than the kernel-space version [7]; (5) Availability and scalability. Zhang developed a mew mechanism named Logic Mirror Ring (LMR) to improve the reliability and availability of the parallel file system. A logic mirror ring is built over all I/O nodes to indicate the mirror relationship among the nodes [8]. (6) Performance evaluation and modeling. The potential impact to application performance in parallel file system environment is not clearly understood and most internal details of the basic components of parallel file system are not public. Therefore, the performance evaluation of parallel file system is of almost importance and the potential performance of parallel file system can be difficult to predict. Currently, some evaluation works have been researched via experimental methods such as [29][30], but these works don't cover some important performance factors such as the number of OSSes, the type of journal for OST, the type of disks, the number of OSS/MDS threads, storage connection approaches etc. Up to now, there is not a suitable model for parallel file system.

Grey system theory was initiated in 1980s by J. Deng in P.R. China [18]. Grey model could be taken as an efficient approximation for extracting system dynamic information and it has various characteristics such as a lower requirement for raw data, it don't need a large amount of raw data and don't require the data on the typical probability distribution[19]. Grey relational analysis (GRA) is an important part of the grey systems theory and it is a quantitative method to explore the similarity and dissimilarity among factors in developing dynamic process and it is the basis of the grey clustering analysis, grey decision-making and grey controlling [20] [21][22].

In this paper, we focus on the performance evaluation and performance modeling using grey system theory. Our experiment evaluation and models cover a lot of critical performance factors.

III. PERFORMANCE MODEL

In this section, we firstly conduct an in-depth survey on the basic performance factors of Lustre file system. Based on this survey, we motivate the work of introducing performance model. A relational analysis model is provided to analyze performance differences of different Lustre systems, which are equipped with different performance factors. Then, we propose a grey prediction model to forecast performance trend when a specific performance factor changes.

A. Survey on Performance Factors

Prior to introducing our model and experiment analysis, we firstly conduct a detailed survey on performance factors of Lustre file system by referring to extensive literatures such as [1],[3],[6], [7], [13], [14], [15], [16]. This part provides some details of performance factors and we categorize these factors as follows: the number of OSSes; the number of OSS/MDS threads; the type of journal for OST; the type of disks; storage connection method; striping pattern (stripe size, stripe count and stripe offset); read/write cache effect; the size and number of inodes for OST/MDT; data distribution strategies etc. The detailed factors distribution and the basic architecture of Lustre can be found in Ref.[31]. Our following performance models mainly focus on some important performance factors: the number of OSSes, the type of journal for OST, the type of disks, the number of OSS/MDS threads and storage connection approaches. Further details on Lustre are available in [9][10][11][12].

B. Relational Analysis Model

Grey relational analysis (GRA) is one of the derived evaluation methods based on the concept of grey relational space (GRS). GRA could be taken as an efficient approximation for extracting system dynamic information and don't require the data on the typical probability distribution.

In this paper, GRA is applied to analyze the relationship of the performance with different factors. Each performance output (i.e. throughput, bandwidth) can be described as a performance vector. Each performance vector can be taken as one series, which consist of a set of criteria or attributes.

Based on similarity and dissimilarity, a relation is the relational measurement of different series (reference series and comparative series). It is a method for determining the relationship between reference data and other comparative data. GRA specifies a mathematical way to analyze the correlation between different series, to determine the "distance" different between a reference series and each of the comparative series.

Assume a reference series $X_0 = (x_0(1), x_0(2), \cdots, x_0(n))$, and m comparative series $X_1 = (x_1(1), x_1(2), \cdots, x_1(n))$,

$X_2 = (x_2(1), x_2(2), \cdots, x_2(n)), \ldots, X_m = (x_m(1), x_m(2), \cdots, x_m(n))$.

The grey relational grades $\gamma(X_0, X_i)$ can be computed by

$$\gamma(X_0, X_i) = \frac{1}{n}\sum_{k=1}^{n} \gamma(x_0(k), x_i(k))$$

$$\gamma(x_0(k), x_i(k)) = \frac{\min\limits_i \min\limits_k |x_0(k) - x_i(k)| + \rho \max\limits_i \max\limits_k |x_0(k) - x_i(k)|}{|x_0(k) - x_i(k)| + \rho \max\limits_i \max\limits_k |x_0(k) - x_i(k)|}$$

where $\gamma(x_0(k), x_i(k))$ denotes grey relational coefficient of performance output with *k*-th factor. $\rho \in [0,1]$ denotes the distinguishing coefficient.

The grey relational grade $\gamma(X_0, X_i)$ is in the interval [0, 1] and grey relational grade can be used to measure the degree of similarity between reference and comparative series. The larger grey relational grade is, the more similar the relationship

between reference series and comparative series is. The fundamental idea of GRA is that the closeness of a relationship is judged based on the similarity level of the geometric patterns of series curves. The more similar the curves are, the higher the GRA degree between series are, and vice versa. GRA satisfies the following four properties: the property of normality; the property of pair symmetry; the property of wholeness and the property of closeness. More details about the four properties can be found in [21] [22] [23].

C. Grey Performance Prediction Model

In our experiments, we discover that the performance series generally meet the exponential distribution, that is, we can simulate the performance of Lustre using some prediction models with the exponential pattern. Xie et. al. indicated that grey prediction model can better fitting the data series with exponential pattern [22]. The exponential pattern of the data series can be obtained by data pre-processing [23]. In this part, we propose an improved residual $GM(1,1)$ model to predict the performance (throughput) changes when a specific performance factor changes, such as the number of threads/OST.

We define the performance series as the grey data series and assume that the original performance series is

$$X^{(0)} = \{x^{(0)}(1), x^{(0)}(2), \cdots x^{(0)}(n)\} \tag{1}$$

In order to obtain better pre-processing data series, we can get the first-order accumulated generating series using 1-AGO (Accumulating Generation Operator), which can transform this original series to satisfy the condition of an exponential pattern. The first-order accumulated generating series is

$$X^{(1)} = \{x^{(1)}(1), x^{(1)}(2), \cdots x^{(1)}(n)\} \tag{2}$$

where

$$x^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i)$$

The AGO can be carried out one time or several times until the series satisfies the condition that all quotients of successive terms $\sigma(k) = \frac{x(k+1)}{x(k)}$ are close to a constant c, where $c \in [e^{-2/[n+1]}, e^{2/[n+1]}]$ [24].

Then, we can obtain the whitenization equation about $X^{(1)}$.

$$\frac{dX^{(1)}}{dt} + aX^{(1)} = u \tag{3}$$

The least-square estimation method is used to determine the solution of parameters a and u in the differential equation and more detailed information about whitenization equation can be found in [18] [19].

By solving equation (3), we get the prediction formula:

$$x^{(1)}(k+1) = [x^{(0)}(1) - \frac{u}{a}]e^{-ak} + \frac{u}{a} \quad (k = 0,1,2....) \tag{4}$$

Because the prediction data of formula (4) is obtained by AGO, IAGO (Inverse AGO) is operated to restore original prediction data.

$$\begin{aligned} x^{(0)}_{GM}(k+1) &= x^{(1)}(k+1) - x^{(1)}(k+1) \\ &= (1-e^{a})(x^{(0)}(1) - \frac{u}{a})e^{-ak} \end{aligned} \tag{5}$$

According to formula (5), we can get prediction data series.

$$X^{(0)}_{GM} = \{x^{(0)}_{GM}(1), x^{(0)}_{GM}(2), \cdots x^{(0)}_{GM}(n)\} \tag{6}$$

Let

$$\xi^{(0)}(k) = x^{(0)}(k) - x^{(0)}_{GM}(k) \tag{7}$$

This whitenization differential fitting has various characteristics such as a lower requirement for raw data, it don't need a large amount of raw data and don't require the data on the typical probability distribution. However, this method requires the data for the modeling is non-negative increments. In the residual analysis often encounter negative item, we can use the abscissa translational approach to tackle this problem.

Let

$$b = \min\{\xi^{(0)}(k)\}, \quad 1 \le k \le n \tag{8}$$

and we can obtain

$$\xi'^{(0)}(k) = \xi^{(0)}(k) + |b| \tag{9}$$

In addition, when making residual series analysis, we often encounter greater dispersion of data. In order to improve the prediction accuracy of the model, we make 20% of the residual series of smoothing.

$$\begin{aligned} &if \quad \xi^{(0)}(k) \succ \bar{\xi}(1+20\%) \; then \; \xi^{(0)}(k) = \bar{\xi}(1+20\%) \\ &if \quad \xi^{(0)}(k) \prec \bar{\xi}(1-20\%) \; then \; \xi^{(0)}(k) = \bar{\xi}(1-20\%) \end{aligned} \tag{10}$$

where

$$\bar{\xi} = \frac{1}{n}\sum_{k=1}^{n} \xi^{(0)}(k)$$

Let $\{\xi'^{(0)}(k)\}$ is the residual series after non-negative operation and 20% smoothing operation and we can construct grey prediction model according to above methods.

$$\begin{aligned} \xi^{(0)}_{GM}(k+1) &= \xi^{(1)}(k+1) - \xi^{(1)}(k+1) \\ &= (1-e^{a})(x^{(0)}(1) - \frac{u}{a})e^{-ak} \end{aligned} \tag{11}$$

By performing the inverse translation transform, original residual series can be obtained.

$$\xi^{(0)}_{GM}(k+1) = \xi^{(0)}_{GM}(k+1) - |b| \tag{12}$$

The final grey prediction model is delivered by fitting formula (5) and formula (11).

$$\hat{x}^{(0)}(k+1) = x_{GM}^{(0)}(k+1) + \xi_{GM}^{(0)}(k+1)$$
$$= (1-e^{a})(x^{(0)}(1) - \frac{u}{a})e^{-ak} + (1-e^{a})(x^{(0)}(1) - \frac{u1}{a1})e^{-ak} - |b| \quad (13)$$

After conducting improved residual $GM(1,1)$ model, we can use this model to predict the performance (throughput) changes when a specific performance factor changes.

IV. Model Analysis

A. Test Cases Design

We perform our performance evaluation tests on 2x Sun Fire X4240 (OSS nodes) which is equipped with 2x AMD Dual-core Opteron™ Processor 3GHz, DDR2 8GB of memory and 24x SAS Disks (300GB), 48x SATA Disks (1TB). The interconnect network is 10 Gigabit TCP/IP network. In software environment, we use Lustre 1.6.6 and Redhat Enterprise Linux 5.2. OBDFilter-survey benchmark is chosen as benchmarking tool.

In our tests, we design 4 groups of cases to reveal the implication of Lustre file system and our tests mainly consider five factors: the number of OSSes (1OSS vs. 2 OSS), the type of journals (internal journal vs. external journal), the type of disks (SAS disk vs. SATA disk), storage connection approaches (directly connected vs. daisy-chain connected) and the number of threads. The detailed groups of cases are design as follows:

Group 1: 1 OSS vs. 2 OSS
case1.1: directly connected OSS server containing 24 SAS disks with external journal from 1 OSS node

case1.2: directly connected OSS server containing 24 SAS disks with external journal from 2 OSS nodes

Group 2: Internal journal vs. External journal
case2.1: directly connected OSS server containing 24 SAS disks with internal journal from 1 OSS node

case2.2: directly connected OSS server containing 24 SAS disks with external journal from 1 OSS node

Group 3: SAS disk vs. SATA disk
case3.1: directly connected OSS server containing 24 SAS disks with external journal from 2 OSS nodes

case3.2: directly connected OSS server containing 24 SATA disks with external journal from 2 OSS node

Group 4: Directly connected vs. Daisy-chain connected
case4.1: directly connected OSS server containing 48 SATA disks with external journal from 2 OSS nodes

case4.2: daisy-chain connected OSS server containing 48 SATA disks with external journal from 2 OSS nodes

B. Model Analysis

In this section, we firstly use the relational analysis model to analyze the characteristics of performance series. Then we apply the grey prediction model to predict the performance (throughput) changes when a specific factor changes.

1) Relational Analysis

In our tests, we choose throughput (GB/Sec) as the basic performance metric. According to the four group cases above, our tests focus on validating the trend of performance changes in general increasing along with the number of threads/OST when different performance factors change. For each group cases, we carry out Write (referred to as W), ReWrite (referred to as ReW) and Read (referred to as R) operations to reveal the performance differences.

Table I to IV are the analysis results ($\rho = 0.5$) of relational analysis model when a specific factor changes, which correspond to group-1-cases, group-2 -cases, group-3-cases and group-4-cases, respectively. Table V to VII show the analysis results when two performance factors simultaneously change.

TABLE I. Relational degree of group-1-cases

γ		Case 1.1			Case 1.2		
		W	**ReW**	**R**	**W**	**ReW**	**R**
Case 1.1	**W**	1	0.987	0.856	0.928	0.93	0.874
	ReW	0.987	1	0.855	0.94	0.942	0.87
	R	0.856	0.855	1	0.865	0.867	0.869
Case 1.2	**W**	0.928	0.94	0.865	1	0.998	0.866
	ReW	0.93	0.942	0.867	0.998	1	0.868
	R	0.874	0.87	0.869	0.866	0.868	1

TABLE II. Relational degree of group-2-cases

γ		Case 2.1			Case 2.2		
		W	**ReW**	**R**	**W**	**ReW**	**R**
Case 2.1	**W**	1	0.982	0.769	0.879	0.906	0.86
	ReW	0.982	1	0.777	0.903	0.892	0.872
	R	0.769	0.777	1	0.77	0.77	0.879
Case 2.2	**W**	0.879	0.903	0.77	1	0.987	0.856
	ReW	0.906	0.892	0.77	0.987	1	0.855
	R	0.86	0.872	0.879	0.856	0.855	1

TABLE III. Relational degree of group-3-cases

γ		Case 3.1			Case 3.2		
		W	**ReW**	**R**	**W**	**ReW**	**R**
Case 3.1	**W**	1	0.998	0.866	0.824	0.836	0.872
	ReW	0.998	1	0.868	0.826	0.838	0.873
	R	0.866	0.868	1	0.808	0.799	0.844
Case 3.2	**W**	0.824	0.826	0.808	1	0.985	0.764
	ReW	0.836	0.838	0.799	0.985	1	0.758
	R	0.872	0.873	0.844	0.764	0.758	1

TABLE IV. Relational degree of group-4-cases

γ		Case 4.1			Case 4.2		
		W	**ReW**	**R**	**W**	**ReW**	**R**
Case 4.1	**W**	1	0.98	0.762	0.963	0.957	0.76
	ReW	0.98	1	0.769	0.983	0.976	0.765
	R	0.762	0.769	1	0.769	0.772	0.96
Case 4.2	**W**	0.963	0.983	0.769	1	0.993	0.765
	ReW	0.957	0.976	0.772	0.993	1	0.768
	R	0.76	0.765	0.96	0.765	0.768	1

TABLE V. Relational degree of case1.2-case2.1

γ		Case 2.1		
		W	**ReW**	**R**
Case 1.2	**W**	0.858	0.802	0.782
	ReW	0.859	0.846	0.782
	R	0.834	0.82	0.849

TABLE VI. RELATIONAL DEGREE OF CASE2.2-CASE3.2

γ		Case 3.2		
		W	**ReW**	**R**
Case 2.2	**W**	0.877	0.87	0.859
	ReW	0.867	0.875	0.858
	R	0.76	0.755	0.969

TABLE VII. RELATIONAL DEGREE OF CASE3.2-CASE4.2

γ		Case 4.2		
		W	**ReW**	**R**
Case 3.2	**W**	0.868	0.864	0.704
	ReW	0.859	0.855	0.699
	R	0.861	0.864	0.869

As can be seen from the Table I to IV, for a specific performance factor, the performance of a pair of read operation has a strong correlation. The performance of rewrite and write operations also have a little closed correlation.

When two factors simultaneously change, the performance of write-write and write-rewrite operations has more closed correlation when we use different types of journals and numbers of OSSes (see Table V), different types of disks and numbers of OSSes (see Table VI) and different types of disks and types of storage connections (see Table VII). However, we also find that the performance of read-read operations has a strong correlation in Table VI. These closed performance correlations of different factors can inspire us to develop some relative performance model like Ref. [27][28].

2) Performance Prediciton

In this part, we focus on the grey prediction model to predict the performance (throughput (GB/Sec)) when the number of threads/OST changes.

In order to better understand our prediction model, we use case1.1-R as the example. In our experiment, the original series of throughput (GB/Sec) of case1.1-R is (0.244, 0.465, 0.648, 0.707, 0.716, 0.829, 0.823, 0.849).

We use the first 6 data as the input data of our model and the latter 2 data as the validation data of the prediction outcome.

Let $X^{(0)} = (0.244, 0.465, 0.648, 0.707, 0.716, 0.829)$

According to Formula (2) to (4), we can obtain

$$x_{GM}^{(0)}(k+1) = 4.348709\exp(0.114612 * k) - 4.104709$$

And we can calculate the residual series.

$$\xi^{(0)} = (0.063097, -0.055771, -0.042853, 0.028799, 0.006247)$$

In order to improve the prediction precise, we conduct the non-negative operation

$$b = \min\{\xi^{(0)}(k)\} = -0.055771$$

$$\xi'^{(0)} = \xi^{(0)} + |b| = (0.118869, 0, 0.012918, 0.08457, 0.062018)$$

The AGO is carried out s until the series satisfies the condition.

$$\xi'^{(1)} = (0.118869, 0.118869, 0.131787, 0.216357, 0.278375)$$

Make 20% of the residual series of smoothing (referred to formula (10))

$$\xi''^{(1)} = (0.118869, 0.118869, 0.131787, 0.1581444, 0.18977328)$$

We can obtain residual prediction formula by Formula (2) to (4).

$$\xi_{GM}^{(0)}(k+1) = 0.648979\exp(0.311641 * k) - 0.53011$$

The final grey prediction model is delivered by fitting operation.

$$\hat{x}^{(0)}(k+1) = x_{GM}^{(0)}(k+1) + \xi_{GM}^{(0)}(k+1)$$
$$= 4.348709\exp(0.114612 * k) + 0.648979\exp(0.311641 * k) - 4.634819$$

Table VIII shows the prediction result of case1.1-R.

TABLE VIII. PREDICTION RESULT OF CASE1.1-R

	x(7)	**x(8)**
Original data	0.823	0.849
Predicted data	0.835	0.917
A.E.	0.012	0.068
R.E.	1.458%	8.009%
A.R.E.	4.734%	

Notes: A.E.=Absolute Error; R.E.=Relative Error;

A.R.E=Average Relative Error

Using the similar way, we can conduct performance prediction using case1.1, case 1.2, case 2.1 and case 2.2. And the average relative errors of prediction results can be found in Table IX:

TABLE IX. AVERAGE RELATIVE ERRORS OF PREDICTION RESULTS

A.R.E.	**W**	**ReW**	**R**
case1.1	9.37%	8.55%	4.73%
case1.2	7.73%	6.38%	5.09%
case2.1	11.31%	10.15%	7.34%
case2.2	8.23%	8.49%	4.09%

Notes: A.R.E=Average Relative Error

As shown in Table IX, our model can get better prediction precision. It has a lower requirement for raw data. It don't need a large amount of raw data and don't require the data on the typical probability distribution. Thereby, these results confirm the previous conclusion which we have obtained from above experiments again, which proves the feasibility of the model.

V. CONCLUSIONS

In this paper, we presented an in-depth efficient performance evaluation of parallel file system based on Lustre file system. At the beginning, we conduct a survey on performance factors which is the foundation of our experiment and model analysis. Then, we propose our relational analysis model to analyze the relationship of performance when we use different factors. We also apply grey prediction model to predict the performance change. In our tests, we design four group cases covering some important performance factors, such as the number of OSSes, storage connection approaches, the type of disks, the type of journal for OST and the number of threads/OST. In our relational analysis, we discover that the performance of write-write and write-rewrite operations has a more closed correlation and the performance of the pair of read

operations also has a strong correlation when we use different type of disks and number of OSSes. Our prediction model can obtain better prediction precision and has many good characteristics such as lower requirement for the mount of raw data and needn't requirement for the data on the typical probability distribution.

ACKNOWLEDGMENT

We thank Verdi March and Atul Vidwansa from Sun Microsystems Inc. for his sincere help, especially in the experiment setup.

REFERENCES

[1] Sun Microsystems, Inc., "LUSTRE™ FILE SYSTEM", Oct. 2008.

[2] F. Wang,Y.L. Yue, D. Feng etc. "High Availability Storage System Based on Two-level Metadata Management", Proceedings of the 2007 Japan-China Joint Workshop on Frontier of Computer Science and Technology(FCST 2007), 2007, pp.41-48.

[3] W. Yu, J. S. Vetter, R. S. Canon etc., "Exploiting Lustre File Joining for Effective Collective IO", CCGrid 2007, 2007.

[4] W. Yu, J. S. Vetter, H. S. Oral, "Performance Characterization and Optimization of Parallel I/O on the Cray XT", IPDPS 2008, 2008.

[5] H. Y. Li, Y. Liu, Q. Cao, "Approximate parameters analysis of a closed fork-join queue model in an object-based storage system", Eighth International Symposium on Optical Storage and 2008 International Workshop on Information Data Storage, 2008.

[6] W. Yu, H. S. Oral, R. S. Canon etc., "Empirical Analysis of a Large-Scale Hierarchical Storage System", Euro-Par 2008, LNCS 5168, 2008, pp. 130-140.

[7] J. Piernas, J. Nieplocha, E. J. Felix, "Evaluation of Active Storage Strategies for the Lustre Parallel File System", SC'07, Nov. 2007.

[8] H. Zhang, W. Wu, X. Dong etc., "A High Availability Mechanism for Parallel File System", APPT 2005, LNCS 3756, 2005, pp. 194-203.

[9] Lawrence Livermore National Laboratory (LLNL), "I/O Guide for LC", Aug. 2007.

[10] Sun Microsystems, Inc., "Solving the HPC I/O Bottleneck: Sun™ Lustre™ Storage System", April 2009.

[11] Sun Microsystems, Inc. and Oak Ridge National Laboratory (ORNL), "Peta-Scale IO with the Lustre File System", Feb. 2008.

[12] DataDirect Networks, Inc., "Best Practices for Architecting a Lustre-based Storage Environment", May 2008.

[13] H. Z. Shan, J. Shalf, "Using IOR to Analyze the I/O performance for HPC Platforms', Cray User Group Conference 2007, Jun. 2007.

[14] W. Yu, S. Oral, J. Vetter etc., "Efficiency Evaluation of Cray XT Parallel IO Stack",Cray User Group Meeting (CUG 2007), 2007.

[15] W. Yu, J. Vetter, "ParColl: Partitioned Collective I/O on the Cray XT", ICPP 2008, 2008.

[16] J. Logan, P. Dickens, "Towards an Understanding of the Performance of MPI-IO in Lustre File Systems", 2008 IEEE International Conference on Cluster Computing, 2008.

[17] Sun Microsystems, Inc., "Lustre™ 1.6 Operations Manual", May 2009.

[18] J. L. Deng, "Control problems of grey systems", Systems and Control Letters, Vol. 1, No. 5, 1982, pp.288-294.

[19] J. L. Deng, "Introduction to Grey System Theory", The Journal of Grey System, Vol. 1, No. 1,1989, pp. 1-24.

[20] J. L. Deng, "Figure on difference information space in grey relational analysis", Journal of Grey System, Vol. 16, No. 2, 2004, pp.96-100.

[21] J. L. Deng, "Grey group decision in grey rationale space", Journal of Grey System, Vol. 10, No. 3, 1998, pp.177-182.

[22] N. M. Xie, S. F. Liu, "Research on evaluations of several grey relational models adapt to grey relational axiom", Journal of Systems Engineering and Electronics, Vol. 20, No. 2, 2009, pp. 304-309.

[23] M. Lu, K. Wevers, "Grey System Theory and Applications: A Way Forward", Journal of Grey System, Vol. 10, No. 1, 2007, pp.47-54.

[24] J. L. Deng, "On judging the admissibility of grey modeling via class ratio", The Journal of Grey System, Vol. 5, No. 4, 1993, pp.249-252.

[25] H. Z. Shan, J. Shalf, "Using IOR to Analyze the I/O performance for HPC Platforms", Cray User Group Conference 2007, 2007.

[26] W. Yu, S. Oral, J. Vetter etc., "Efficiency Evaluation of Cray XT Parallel IO Stack", Cray User Group Meeting (CUG 2007), 2007.

[27] M. P. Mesnier, M. Wachs, R. R. Sambasivan, "Modeling the Relative Fitness of Storage", SIGMETRICS'07, 2007, pp.37-48.

[28] M. P. Mesnier, M.Wachs, R. R. Sambasivan etc. "Relative Fitness Modeling", Communications of the ACM, Vol.52, No. 4, 2009, pp.91-96.

[29] W. Yu, H.S. Oral, J. Vetter. "Efficiency Evaluation of Cray XT Parallel IO Stack.", Cray User Group Meeting (CUG 2007), 2007.

[30] H. Shan, J. Shalf. "Using IOR to Analyze the I/O performance for HPC Platforms", Cray User Group Conference 2007, 2007.

[31] T. Zhao, V. March, S. Dong et al. "Evaluation of A Performance Model of Lustre File System". Proceeding of the Fifth Annual ChinaGrid Conference, 2010,pp.191-196