I/O Response Time in a Fault-Tolerant Parallel Virtual File System

Dan Feng[1], Hong Jiang[2], Yifeng Zhu[2]

[1] Key Laboratory of Data Storage System, Ministry of Education
College of Computer, Huazhong University of Science and Technology, Wuhan, China
dfeng@hust.edu.cn
[2] Department of Computer Science and Engineering
University of Nebraska – Lincoln, Lincoln, Nebraska
Jiang@csce.unl.edu

Abstract. A fault tolerant parallel virtual file system is designed and implemented to provide high I/O performance and high reliability. A queuing model is used to analyze in detail the average response time when multiple clients access the system. The results show that I/O response time is with a function of several operational parameters. It decreases with the increase in I/O buffer hit rate for read requests, write buffer size for write requests and number of server nodes in the parallel file system, while higher I/O requests arrival rate increases I/O response time.

1 Introduction

Parallel Virtual File System (PVFS) [1] is a parallel file system for Linux clusters, which stripes the data among the cluster nodes and accesses these nodes in parallel to achieve high I/O throughputs. A Cost-Effective Fault-Tolerant Parallel Virtual File System (CEFT-PVFS) [2] has been designed and implemented to meet the critical demands on reliability while still being able to deliver a considerably high throughput.

When multiple clients submit data-intensive jobs at the same time, the response time experienced by the user is an indicator of the power of the cluster. In this paper, a queuing model is used to analyze in detail the average response time when multiple clients access the fault tolerant parallel virtual file system.

2 Architecture of the Fault Tolerant Parallel Virtual File System

The diagram of the fault tolerant parallel virtual file system is shown in Fig.1. All I/O server nodes are divided into two groups, the primary one and the mirroring one. File data is striped across the primary group and duplicate on the mirroring group. When Writing, data are stored in the primary group in RAID 0 style and backed up in the mirroring group simultaneously. Data is retrieved from the nodes that have less workload between the mirrored pair to optimize the read performance.

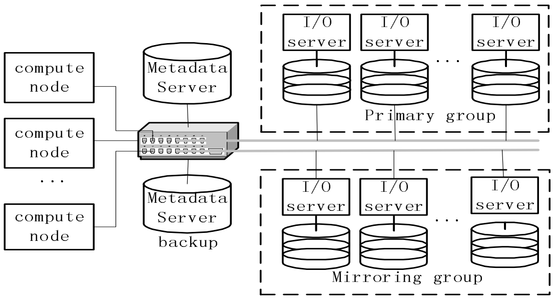

Fig.1. Diagram of the fault tolerant parallel virtual file system

3 Definitions and Notations of Disk I/O Parameters

I/O response time in the fault tolerant parallel virtual file system depends primarily on the network bandwidth, I/O buffer size and policy, and disk I/O service time. The disk I/O may become the bottleneck and it is determined by three main parameters, namely, seek time, rotational latency and transfer rate.

Definitions and notations for several relevant disk I/O parameters are given below:

C: Number of disk cylinder;

S: disk seek time, with its maximum being denoted by $S_{\max}$;

R: disk rotational latency, with the full rotation time being denoted by $R_{\max}$;

D: disk seek distance, which is a random variable in the range of $[0, C-1]$;

p_s: the probability that the seek distance is 0, $P_s = P\{D=0\}$;

L: data striping size;

T_d : data transfer time, $T_d = L / r_d$, where r_d is disk transfer rate;

Y_r, Y_w, Y : disk read, write and whole service time, respectively.

The relationship between disk seek time and the seek distance *i* is given as:

$$S \approx a + b\sqrt{i} \qquad i > 0$$

where S is the seek time, *a* is the arm acceleration time, *b* is the factor of seeking track. *a* is the seek time between two neighboring cylinders.

The mean seek time and the second moment can be accurately approximated by[4]

$$E(S) = (1-p_s)[a + \frac{8}{15} b\sqrt{C-1}]$$

$$E(S^2) = (1-p_s)[a^2 + \frac{1}{3} b^2 (C-1) + \frac{16}{15} ab\sqrt{C-1}]$$

When requests to a disk are independent of one another, the rotation time is assumed to be uniformly distributed in $[0, R_{\max}]$, with probability density function:

$$f_R(x) = 1 / R_{\max} \qquad 0 \le x \le R_{\max}$$

The mean rotation time and the second moment are:

$E(R) = \frac{1}{2} R_{\max}$

$E(R^2) = \frac{1}{3} R_{\max}{}^2$

The disk drive service time is $Y = S + R + T_d$.

M.Y.Kim studied traces of the real disk service times and found it to be generally distributed [5]. Therefore, we adopt the M/G/1 model to analyze the disk response time in the cluster.

4 I/O Response Time Analysis

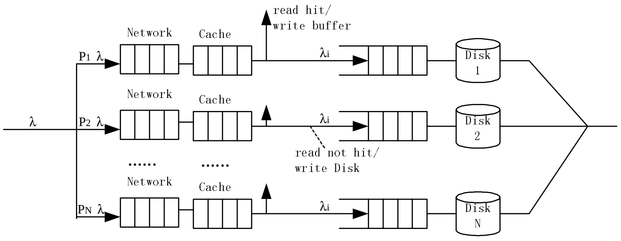

Fig.2. The Queuing Model of I/O Service

The queuing model for the system service under data-intensive load is shown in Fig.2. Part of the main memory space in a server node is used for I/O cache buffer to hide the disk I/O latency and to take advantage of data reference locality. Assume the number of I/O server nodes to be N in each group. I/O requests follow a Poisson process; with a mean arrival rate of λ . The arrival rate to the server node *i* is $P_i\lambda$, where P_i is the probability that the request is directed to node *i*. When the I/O request is a small read or write, where the data size is equal to or less than the size of a striped block, and the workload on a server node among a group is balanced, P_i is equal to 1/N. When the I/O request is a large read or write, where data is striped on all of the nodes in a group, P_i is equal to 1. So the typical range of P_i is [1/N, 1]. Let P_r and $P_w = 1 - P_r$ denote the read and write probability of a request, respectively. Assume the I/O cache buffer hit rate for read requests to be h_r and the probability of write buffer being full to be f_w . Thus, the effective arrival rate to each disk is: $\lambda_i = [P_r(1-h_r) + P_w f_w] \cdot P_i\lambda$.

Assume that the network service time and the I/O buffer service time are exponentially distributed with the average times T_{net} and T_c , respectively. Therefore, request residence time W_{net} in the network and W_{Cache} in the I/O buffer can be modeled using the M/M/1 queuing model [6]. The average residence time W_{disk} in a disk drive can be calculated according to the M/G/1 model.

The average I/O response time can be expressed as:

$Z = W_{net} + W_{cache} + P_{disk} \cdot W_{disk}$

$$= \frac{T_{net}}{1 - P_i \lambda \cdot T_{net}} + \frac{T_c}{1 - P_i \lambda \cdot T_c} + [P_r(1 - h_r) + P_w f_w] \times \{ \frac{\lambda_i E(Y^2)}{2[1 - \lambda_i E(Y)]} + E(Y) \}$$

Where $T_{net} = L / R_{net}, T_c = L / R_{memory}$ and L is the size of data to be accessed on each I/O server node. Even though the data is striped in fixed blocks to server nodes in a RAID0 style, the blocks can be incorporated into a large block with length equal to L. R_{net} and R_{memory} is the available network bandwidth and the available memory access rate respectively.

The average I/O response time can be obtained from above formula for Z under different workload and application environments with different parameters.

5 Conclusion

The I/O response time in the fault tolerant parallel virtual file system is discussed in the paper. The analytical results show the different level of sensitivity of the average I/O response time to various system and operational parameters such as I/O buffer size, data locality, read/write probability, request data size and server group size. These results provide useful insight into the complicated relationships among different system and operational parameters, thus allowing potential optimization for system configuration and I/O performance.

6 Acknowledgment

This work was partially supported by National Science Foundation of China No.60273074, NSF grant (EPS-0091900) and Huo Yingdong Education Foundation.

References

1. P. H. Carns, W. B. Ligon III, R. B. Ross, and R. Thakur: PVFS: A Parallel File System For Linux Clusters, Proceedings of the 4th Annual Linux Showcase and Conference, Atlanta, GA (2000) 317-327
2. Yifeng Zhu, Hong Jiang, Xiao Qin, Dan Feng, David R. Swanson: Design, Implementation, and Performance Evaluation of a Cost-Effective Fault-Tolerant Parallel Virtual File System. Int. Workshop on Storage Network Architecture and Parallel I/Os, New Orleans, LA (2003)
3. Kai Hwang, Hai Jin, Edward Chow, Cho-Li Wang, and Zhiwei Xu: Designing SSI Clusters with Hierarchical Checkpointing and Single I/O Space. IEEE Concurrency, IEEE Computer Society Press, Vol.7, No.1,(1999) 60-69
4. S. Chen and D. Towsley: The Design and Evaluation of RAID 5 and Parity Striping Disk Array Architectures. Journal of Parallel and Distributed Computing, Vol. 17, (1993)
5. M.Y.Kim and A.N.Tantawi: Asynchronous Disk Interleaving: Approximating Access Delays. IEEE Trans. on Computer.(1991)
6. L.Kleinrock: Queueing System. Vol.1, John Wiley, New York (1975)