

Performance Effect Analysis of False Sharing Problem in Clusters with Single I/O Space

Hai Jin and Kai Hwang

University of Southern California
Los Angeles, California, 90089

Abstract : With the development of cluster of workstations, more and more researches are focused on the single system image (SSI). Single I/O space plays an important role in the I/O intensive applications. Based on the study of the different I/O architectures of cluster, false sharing problem in the distributed RAID with single I/O space is arisen.

Identification of false sharing problem plays an important role for the performance improvement of the storage system in the cluster environment. We give out the precise definition of different cases of false sharing. In order to evaluate the false sharing effect to the I/O system performance, we define a performance measurement method. Based on the precise definition of false sharing and the measurement metric, simulation is carried out to test the false sharing effect to the overall I/O system performance. From the simulation result, we can see that because of the false sharing effect, the overall I/O system performance will not increase with the increasing of the stripe unit size. This is quite different from the scenario of centralized disk array.

Keywords: Clusters of workstations, False sharing, Storage system architecture, Single I/O space, Performance analysis

1. Introduction

The severe bottleneck problem between processor speed and disk bandwidth has resulted in a configuration of multiple disks known as *Redundant Arrays of Independent Disks* (RAID) [4][6]. They are capable of providing improved levels of reliability, availability, and performance over single disk.

On the other hand, if a site fails permanently because of flood, earthquake or other disaster, then a RAID will also failed. That is to say, a RAID offers no assistance with site disasters. Moreover, if a site fails temporarily, because of a power outage, a hardware or software failure, etc., then the data on a RAID will be unavailable for the duration of the outage. Hence, some researches begin to extend the RAID concept to a distributed computing system, especially with the widely used of *network of workstation* (NOW) and cluster. Research on the distributed RAID in clusters is becoming more and more important. It at least has following main goals and objectives.

- Distributed RAID in clusters can provide higher availability, especially for the site disaster tolerance.
- Distributed RAID in cluster can provide parallel I/O functionality. This is very important to the I/O intensive applications, such as multimedia applications and I/O

intensive massive parallel computing applications [3][19]. More and more researches on parallel I/O are carried on a virtual shared memory on top of a *distributed shared memory* (DSM). Workstation cluster offers a single I/O space on top of distributed-memory hardware architecture.

- Research on distributed RAID in clusters can provide through understanding and implementation of single I/O space, which is one of the important *single system image* (SSI) characteristics of cluster. Single I/O space implies that any node in the cluster can access the storage systems even if they are attached to the different nodes.

In this paper, we will concentrate on the topic of false sharing problems of distributed RAID in clusters with single I/O space. The paper is structured as follows. In section 2, we will briefly discuss the different distributed RAID architecture in the cluster of workstations with single I/O space. Section 3 will give a precise metric definition of performance effect of false sharing problem. Section 4 gives out the detail description of the simulation of false sharing effect to the overall I/O system performance. Finally, section 5 closes with conclusions and mentions several research topics for future research.

2. Distributed RAID Architectures in Cluster with Single I/O Space

Three I/O architecture design options are assessed below for enhancing the availability and fault tolerance of a cluster of workstations or PCs.

M. Stonebraker and G. Schloss first proposed the RADD (*Redundant Arrays of Distributed Disks*) architecture [22] as a multicopy algorithm for distributed RAID systems. All local disks, attached to different cluster hosts, logically form the RADD subsystem. Normally, it stores the checkpointing data in local disk blocks sequentially while parity blocks reside in other local disks.

Among different nodes, the RADD applies the RAID-5 algorithm to handle local I/O operations, which are transparent to higher-level RADD operations. For simplicity, you can readily apply the RAID-1 architecture on local disks. RADD implements mirroring on neighboring disks, but there is no parity among the distributed local disks.

In *network-attached secure disks* (NASD) [7][8], the RAIDs are directly attached to the network as a stable storage to allow shared access by all cluster nodes. Each workstation node in the cluster may or may not have local disk attached. Even with locally attached disks, they serve to buffer the data retrieved from the NASD to local nodes. NASD supports independent accesses by all cluster nodes.

The NASD architecture is quite different from the server-attached RAID. Data blocks transfer directly from the network to the end users at local workstations instead of through the network server. The NASD improves the scalability by removing the bottleneck problem on the network server.

The third I/O architecture combines the advantages of both earlier architectures for better support of fault tolerance in case of single or multiple failures in cluster hosts or local disks. It is conceptually illustrated in Fig.1 [9]. The cluster nodes or hosts are either workstations or PCs. All nodes are connected by Gigabit LAN or SAN. Local disks are attached to each workstation node. Each local disk is accessible from its own host attached as depicted by the vertical arrows in Fig.1. All the local disks form an RADD as described above. The network-attached RAIDs form a NASD to be used as the stable storage for better support of fault tolerance. All the local disks and NASD form a Single I/O Space.

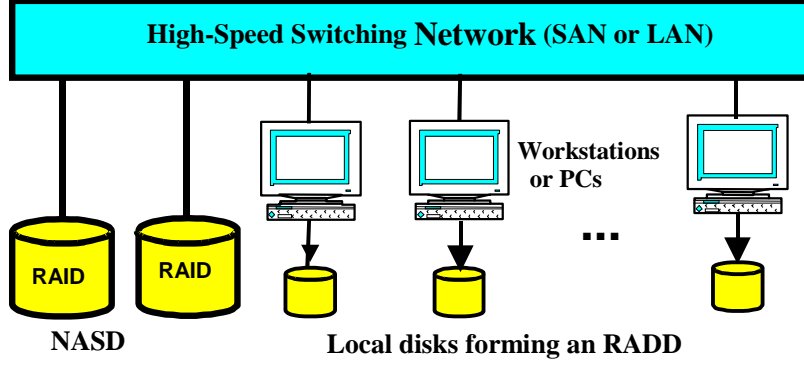


Figure 1 I/O architecture of workstations cluster built with RADD and NASD

3. Measurement of False Sharing Effect

In single I/O space, data are shared at the block level. False sharing has been recognized as a major source of memory inconsistency in any *distributed shared memory* [1][2][15] systems. False sharing problem in single I/O space may occur at both data blocks and at parity blocks [12].

To access data in the disk array, the minimum access unit is block. If different processes access the different part of the block, there is no true sharing between them. This is called data false sharing in single I/O space [12]. The main source of data false sharing is the access of several fragments in a block. The size of the block is about one track of the disk drive. It is very common that a file can not occupy the entire space of a block.

Parity block in single I/O space is shared by all hosts. For each *write* operation, the parity should be modified to keep the memory consistency. But sometimes, parity is falsely shared. In the case of data fragmentation, each process writes to a different fragment of the data block. This triggers the modification of different fragments of the parity block as well. Thus data false sharing will trigger parity false sharing [12].

We define the false sharing measurement based on the basic unit of disk access. Different from memory reference where the basic reference unit is word, the basic unit of disk access is sector. The physical size of a sector is 512 bytes, while the size of logical sector depends on the different file system. The term sector hereafter refers to logical sector. We use the term *sector* to refer to all atomic disk access, regardless of their actual size.

The coherent unit in the distributed RAID is a block (b), which is a set of sectors:

$$b_j = \{ s_i | \text{sector } i \text{ is part of block } j \} \quad b_j \cap b_k = \emptyset, j \neq k \quad (1)$$

The number of disk accesses made to the sector s_i is denoted as $s_{i,a}$ and the number of write accesses is denoted as $s_{i,w}$. Similarly, we denote the total number of accesses to the block b_j as $b_{j,a}$ and the number of write accesses to the block as $b_{j,w}$. All of these counts are taken over the interval of time of interest.

The *processor set* (PS) of a sector is defined as the set of processors that access the sector over the time interval of interest:

$$S_i = \{ \text{processors accessing } s_i \} \quad (2)$$

The PS of a block is defined as the union of the processor sets of the sectors in the block:

$$B_j = \bigcup_{s_i \in b_j} S_i = \{ \text{processors access } b_j \} \quad (3)$$

Using the above definitions, an expression for the false sharing that can be attributed to a particular s_i in b_j can be illustrated as follow:

$$F(i, j) = 1 - \frac{|S_i|}{|B_j|} \quad (4)$$

The above definition indicates that the greater the difference between the sector's PS size and the block's PS size, the greater the degree of false sharing associated with that sector. But this measurement does not give the precious indication of false sharing effect to the system performance. In order to specify the false sharing effect to the distributed RAID system performance, we define the term of *one-processor-sector set* (OPSS), B_j^1 . OPSS is the set of sectors within a certain block b_j , which is accessed by only one processor over a time interval of interest. That is:

$$B_j^1 = \bigcup_{s_i \in b_j} \{S_i \mid |S_i| = 1\} = \bigcup_{s_i \in b_j} \{\text{only one processor access } s_i \text{ in } b_j\} \quad (5)$$

Thus, the expression for the false sharing associated with a particular b_j can be illustrated as follow:

$$F(j) = \frac{|B_j^1|}{|B_j|} \quad (6)$$

As for most cache coherence scheme, the primary cause of coherence overhead is due to write reference [9]. With the invalidation-based protocol, writes cause the invalidation that in turn can cause false sharing misses, and with the updated-based protocol, it is the write references that cause false sharing updates. We use $B_{j,a}^1$ to indicate the set of sectors within a certain block b_j accessed by only one processor over a time interval of interest, while $B_{j,w}^1$ to indicate the set of sectors within a certain block b_j written by only one processor. Thus, the weight $F(j)$ can be expressed as:

$$F_w(j) = F(j) \times \frac{B_{j,w}^1}{B_{j,a}^1} \quad (7)$$

Thus, sectors used in a mostly read-only fashion will have $F_w(j)$ close to zero, while for the sectors with high write-to-access ratios will have values closer to $F(j)$.

4. Simulation and Performance Analysis

In order to evaluate the performance effect of false sharing in the distributed RAID, we carry the following simulation using the simulation tools designed by ourselves. We will give the detail of the simulation methodology and the workload characteristics in this section. We will also give the simulation results based on our testbed by using the above measurement method. Detailed analysis of the simulation results will be given at the end of this section.

4.1 Simulation Methodology

In order to evaluate the performance effect of false sharing in the distributed RAID, we need a testbed in which we can easily modify architectural parameters of distributed RAID and collect I/O access traces. We also need the I/O trace accurately measure the false sharing in the

distributed RAID.

The testbed has the ability of reconfiguring as any architecture of RAID level 0, 1, 4, 5 and 10. We can easily change the architecture parameters, such as stripe unit size. We use the network-attached disk array as the architecture simulation model of distributed RAID. That is, we connect the disk array to the network and access it via different site in the network. We configure the disk array with two strings with two 540MB Quantum disk drivers per string. We use 0 and 1 to represent two different strings and use 1 and 2 to represent disk drivers in each string. Therefore, we denote disk drivers in the system as $D[0,1]$, $D[0,2]$, $D[1,1]$ and $D[1,2]$, respectively. We configure these four disk drivers as one single parity group with RAID level 5 in our following simulation.

We design a program to simulate the behavior of the disk array in the cluster. We also develop a set of synthetic workloads as the workload of distributed RAID. The generation of workload is by using the I/O trace of Qbench, which is a disk driver benchmark designed by Quantum Co. to measure the disk service time and the channel interaction time of single disk driver as well as storage system.

For the type of I/O access, the percentage of read and write can be set flexible. In order to collect as many as I/O trace as the input of our simulation program, we can ignore the read access because read access has no contribution to the study of data sharing. In this way, we get the I/O trace and use it as the input of our simulation program to investigate the performance effect of false sharing in the distributed RAID.

4.2 Simulation Results and Analysis of Performance Effect of False Sharing

We use the I/O trace of Qbench as the input of our simulation program. We configure the architecture of disk array to RAID level 5. We select the stripe unit size of disk array as 2KB, 4KB, 8KB, 16KB, 32KB, respectively. In the simulation, we use two processes, each with the synthetic workloads from Qbench I/O trace. For one process, the I/O access size each time is one sector, and for another process, the I/O access size each time is two sectors.

Figure 2 shows the performance effect of false sharing during the I/O accesses between these two processes with different stripe unit size. In these figures, x -coordinate shows the different stripe unit size and the different disk drivers in the disk array. y -coordinate shows the false sharing effect by using the measurement metric we defined in this paper. That is, the percentage of OPSS to the PS size.

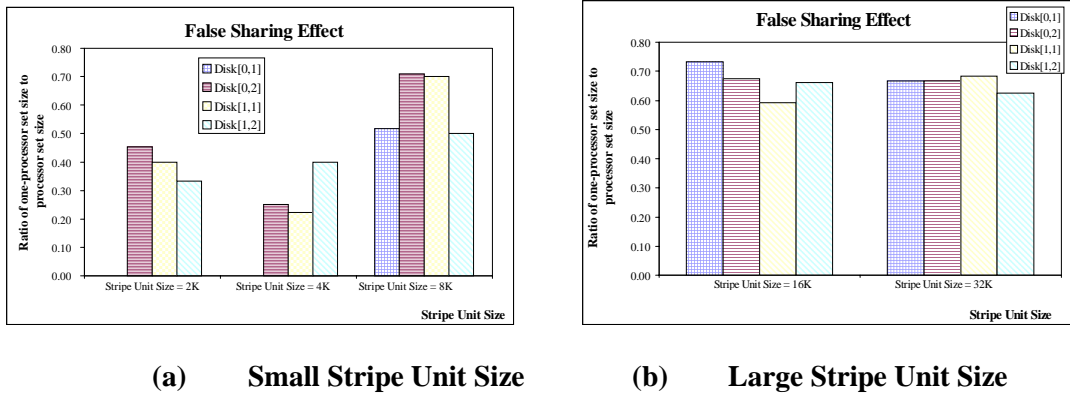


Figure 2 False Sharing Effect using OPSS to PS size as measurement metric

Figure 3 uses another metric to measure the performance effect of false sharing. That is,

y-coordinate shows the percentage of false sharing I/O access times between two processes to the total I/O access times for each disk driver in the different stripe unit size environment.

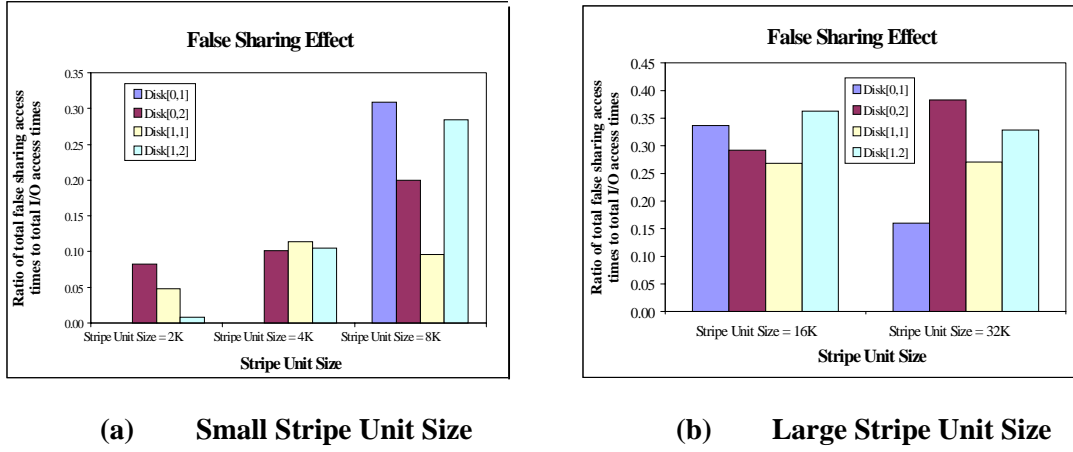


Figure 3 False Sharing Effect using total false sharing access times to total I/O access times as measurement metric

From these figures, we can see that the larger the stripe unit size, the heavy case of false sharing. This is quite different from the situation which disk array connects to the single machine. In the case of cluster or network environment, it is not the case that the large the stripe unit size, the better the performance. This is because the large the stripe unit size, the large the data block will be cached for each processor. According to the research of [23], the false sharing phenomena will be severer when the cached data block achieves to certain size.

For traditional disk array control method, the fixed amount of data with size of stripe unit will be prefetched to each buffer for each disk. If the I/O access is just the full stripe unit access, there is no false sharing. If the I/O access is not the full stripe unit access, it may occurs the false sharing with other processors according to our precise definition of data false sharing and parity false sharing in the distributed RAID. The source of data false sharing is the fragment access of stripe unit that is prefetched to the buffer. In order to mitigate the false sharing, an efficient disk array control method should be developed.

In [12], we propose an efficient parallel I/O control method, called *LCR (Length-variable Command Recombination)*, to improve the performance of disk array, it is especially benefit to the distributed RAID to reduce the false sharing effect. The basic idea of LCR is by using *command recombination* technique to reduce the number of sub I/O commands to each disk drivers in the array. When the data length is longer than the product of stripe unit size and the number of member disks in the disk array, combining the sub I/O commands which belongs to the same disk into single sub I/O command. The data length of the combined sub I/O command is the sum of all of the data length of former sub I/O commands.

Using LCR method can reduce the I/O operations to the disk array to some extend to improve the efficiency of I/O operation in the disk array. To the benefit of reduce the false sharing effect, using LCR method will not increase the irrelevant data content in each buffer, so that all the data fetched in to the buffer are accessed by disk driver, without having the situation of fragment access of stripe block.

5. Conclusions

With the application of network of workstations or clusters, more and more I/O intensive applications have the higher requirement of the storage system of cluster to meet the high bandwidth and high availability requirements. Distributed RAID with single I/O space is a new topic of the research in this area. The main aim of the distributed RAID is to provide high availability to the cluster system, even if in the case that one site has suffered disaster, data in this site can still be recovered. The information used for checkpoint recovery can be recovered to maintain the system working without interrupted.

The main research topic in this paper is the performance effect of false sharing problem in the distributed RAID. The false sharing phenomena in distributed RAID can mainly classified into two different cases, parity false sharing and data false sharing. We give a mathematical calculation method to evaluate the effect of false sharing in the distributed RAID. We use the one-processor set size (OPSS) to the processor set (PS) size as the metric to measure the false sharing effect to the overall system I/O performance. This method to measure the false sharing effect to the I/O performance is much more precise and easy understanding.

We carry on our research on our experimental platform by using the I/O trace as the input of different processor in the cluster-based environment to evaluate the false sharing effect to the I/O system performance. From the simulation results, we find that the I/O performance of distributed RAID is quite different from traditional disk array system. The larger the stripe unit size, the I/O performance will not always increase. The false sharing effect will increase after the stripe unit size achieves to a certain point. This is because the larger the stripe unit size, the higher frequency the fragment access to the stripe unit which is cached to the buffer of each disk driver. In order to reduce the false sharing effect to the I/O system performance, an efficient disk array control method should be developed to meet the requirement of distributed RAID. LCR method [12] is an efficient control method to improve the performance of distributed RAID system.

References

- [1] . C. Amza, A. L. Cox, S. D. Warkadas, P. Kelehr, H. Lu, R. Rajamony, W. Yu and W. Zwaenepoel, "TreadMarks: Shared Memory Computing on Networks of Workstations", *IEEE Computer*, Vol.29, No.2, 1996, pp.18-28
- [2] . W. J. Bolosky and M. L. Scott, "False Sharing and its Effect on Shared Memory Performance", *Proceedings of the USENIX Symposium on Experiences with Distributed and Multiprocessor Systems (SEDMS IV)*, Sept. 1993, pp.57-72
- [3] . P. Brezany, *Input/Output Intensive Massively Parallel Computing: Language Support, Automatic Parallelization, Advanced Optimization, and Runtime Systems*, Lecture Notes in Computer Science, Vol.1220, Springer Verlag, 1997
- [4] . P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz and D. A. Patterson, "RAID: High-Performance, Reliable Secondary Storage", *ACM Computing Surveys*, Vol.26, No.2, June 1994, pp.145-185
- [5] . S. J. Eggers and T. E. Jeremiassen, "Eliminating False Sharing", *Proceedings of the 1991 International Conference on Parallel Processing*, August 1991, Vol.1, pp.377-381
- [6] . G. Gibson; *Redundant Disk Arrays: Reliable, Parallel Secondary Storage*, MIT Press, 1992
- [7] . G. A. Gibson, D. F. Nagle, K. Amiri, F. W. Chang, E. M. Feinberg, H. Gobioff, C. Lee, B. Ozceri, E. Riedel, D. Rochberg and J. Zelenka, "File Server Scaling with Network-Attached Secure Disks", *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (Sigmetrics '97)*, June 1997,

- [8] . G. A. Gibson, D. F. Nagle, K. Amiri, F. W. Chang, E. M. Feinberg, H. Gobioff, C. Lee, B. Ozceri, E. Riedel and D. Rochberg, "A Case for Network-Attached Secure Disks", CMU SCS technical report CMU-CS-96-142, September 1996
- [9] K. Hwang, H. Jin, E. Chow, C.-L. Wang, Z. Xu, "Designing SSI Clusters with Hierarchical Checkpointing and Single I/O Space", *IEEE Concurrency*, Vol.7, No.1, 1999, pp.60-69
- [10] K. Hwang, Z. Xu; *Scalable Parallel Computing: Technology, Architecture, programming*, WCB/McGraw-Hill Co., 1998
- [11] R. L. Hyde and B. D. Fleisch, "Degenerate Sharing", *Proceedings of the 1994 International Conference on Parallel Processing*, 1994, Vol.1, pp.267-270
- [12] Hai Jin, Jin He, Qiong Chen, and Kai Hwang, "Grouped RAID Accesses to Reduce False Sharing Effect in Clusters with Single I/O Space", *Proceedings of International Symposium on High Performance Computing '99*, May 26-28, 1999, Kyoto, Japan
- [13] H. Jin and K. Hwang, "False Sharing Problems in Distributed RAIDs", *Proceedings of 1999 ACM Symposium on Applied Computing*, February 28 – March 2, 1999, San Antonio, Texas, USA
- [14] V. Khera, R. P. Larowe and C. S. Ellis, "An Architecture-Independent Analysis of False Sharing", *Computer Science Department, Duke University, TR 93-006*, Oct. 1993
- [15] V. Khera, "Factor Affecting False Sharing on Page-Granularity Cache-Coherent Shared-Memory Multiprocessors", *Computer Science Department, Duke University, CS-1994-37*, Dec. 1994
- [16] A. N. Mourad, W. K. Fuchs, and D. G. Sbba, "Assigning Sites to Redundant Clusters in a Distributed Storage System", *Proceedings of 1993 International Conference on Parallel Processing*, Vol.1, pp.64-71
- [17] S. Nakamura, H. Minemura, T. Yamaguchi, H. Shimizu, T. Watanabe and T. Mizuno, "Distributed RAID Style Video Server", *IEICE Transactions on Communication*, Vol.E79-B, No.8, August 1996, pp.1030-1037
- [18] G. F. Pfister, *In Search of Cluster: The Ongoing Battle in Lowly Parallel Computing*. Second Edition, Prentice-Hall PTR, 1998
- [19] E. Riedel and G. A. Gibson, "Understanding Customer Dissatisfaction With Underutilized Distributed File Servers", *Proceedings of the Fifth NASA Goddard Space Flight Center Conference on Mass Storage Systems and Technologies*, September 1996
- [20] E. Riedel and G. A. Gibson, "Active Disks: Remote Execution for Network-Attached Storage", *CMU SCS technical report CMU-CS-97-198*, December 1997
- [21] E. Riedel, G. A. Gibson and C. Faloutsos, "Active Storage For Large-scale Data Mining and Multimedia", *Proceedings of the 24th International Conference on Very large Databases (VLDB'98)*, August 1998
- [22] M. Stonebraker and G. A. Schloss, "Distributed RAID – a New Multiple Copy Algorithm", *Proceedings of the Sixth International Conference on Data Engineering*, Feb. 1990, pp.430-437
- [23] J. Torrellas, M. S. Lam and J. L. Hennessy, "False Sharing and Spatial Locality in Multiprocessor Caches", *IEEE Transactions on Computers*, Vol.C-43, No.6, June 1994, pp.651-663