DRALIC: A Peer-to-Peer Storage Architecture

Xubin He, Ming Zhang, and Qing Yang
Dept. of Electrical and Computer Engineering
University of Rhode Island
Kingston, RI 02881

Abstract *The ideal storage system should provide high availability, reliability, and unlimited performance and capacity with minimal management. This paper describes the architecture and performance of DRALIC—Distributed RAid and Location Independent Caching. DRALIC is a peer-to-peer storage architecture that attempts to make this ideal storage into practice. The main idea of DRALIC is to combine or bridge the disk controller and network controller of existing PCs interconnected by a high-speed LAN switch. To demonstrate our approach, we have implemented a simulator called DralicSim based on a set of PCs running Windows NT. Preliminary performance measurements suggest that our architecture achieves its goal.*

Key Words: Peer-to-Peer Storage, Disk I/O, Cluster, Distributed RAID

1. Introduction

Emerging high-speed networks allow machines to access remote data nearly as quickly as (even faster than) they can access local data [1,2,9,10]. We design and evaluate a new architecture called *DRALIC*—Distributed RAid and Location Independence Caching. *DRALIC* provides a peer-to-peer [5], direct and immediate solution to boost web server performance by making use of commodity computers that are available today. *DRALIC* starts working only when an actual disk request has come to the device no matter whether it is a result of file system miss or it is a request from a database operation. It does not require any change of existing operating systems, databases, nor applications. In one implementation, *DRALIC* combines the functions of disk I/O host bus adapter card (HBA) and the functions of the network interface card (NIC) to form an integrated I/O-Network card with a highly intelligent embedded-processor. Or in another implementation, *DRALIC* bridges the HBA and NIC by designing intelligent device drivers. Besides network accesses, the new interface card or drivers at each node control the local disk as well as a raw RAM partition of the system RAM of the node. The disk together with the ones in other nodes in the network forms a distributed RAID [3,7] that appears to users as a large and reliable logic disk space. The raw RAM partitions in all nodes together form a large, global, and location independence cache for the RAID and is accessible to any node connected to the network, independent of its physical location. Therefore, *DRALIC* works at device or device driver level to allow all the nodes to work together in parallel to process web requests. The distributed RAID allows parallel operations of disk accesses and provides fault tolerance using parity disks, whereas location independence caches

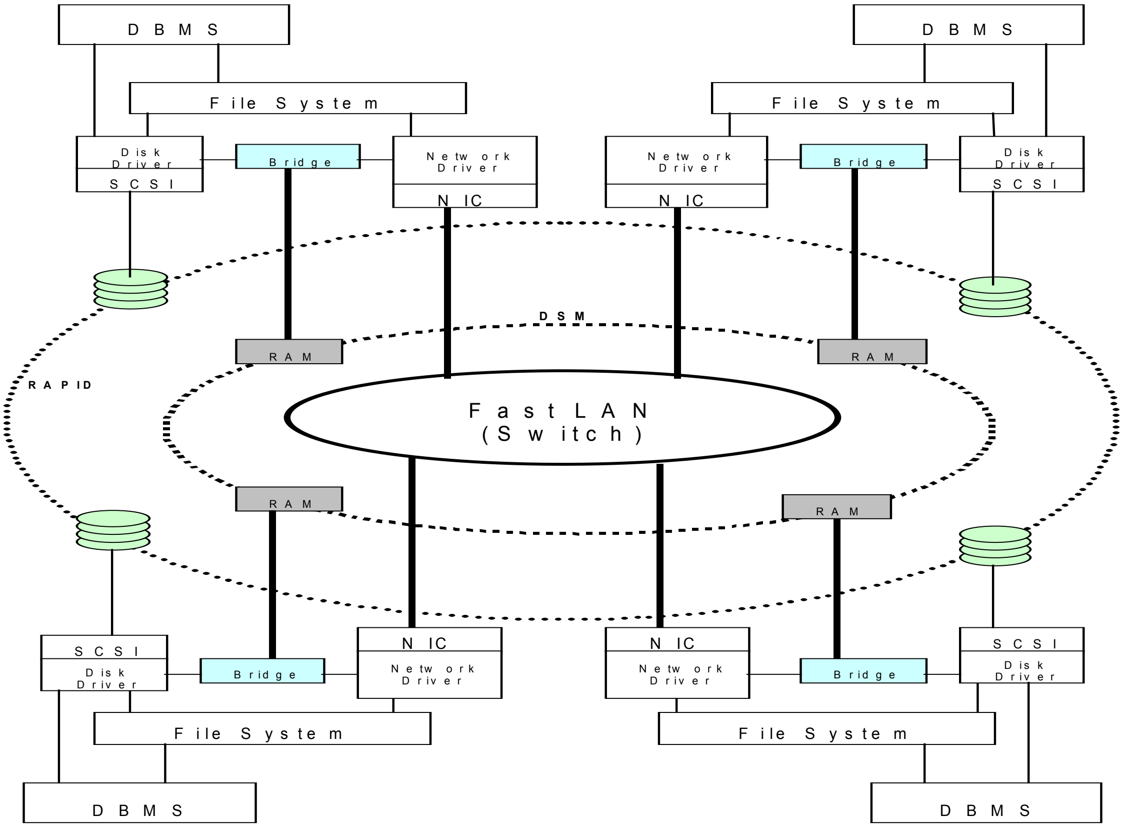


Figure 1: DRALIC Architecture (In this scenario, 4 PCs are connected through the switch, and the DRALIC bridge combines the HBA and NIC within each PC. RAMs from each PC form a shared memory and DRALIC Disks from each PC are organized as distributed disk array.)

provide cooperative caching to the computing nodes for better I/O performance. Furthermore, *DRALIC* is a cost-effective architectural approach because it uses low cost PCs/Workstations that are often readily available as existing computing facilities in an organization or cooperation.

2. DRALIC Architecture

The main idea of *DRALIC* is very simple. It combines or bridges disk I/O host bus adapter card (HBA) and network interface card (NIC) to implement distributed RAID and global caching. Figure 1 shows the conceptual diagram of a *DRALIC*. A disk that exists in a PC/Workstation (node) is partitioned into two parts: one local disk that holds OS and local data and applications, and the other called *DRALIC* disk that is used by *DRALIC*. *DRALIC* disks in all nodes in the system are interconnected through the *DRALIC* controller and a network switch to form a distributed RAID. The system RAM in each node is also partitioned into two parts: one is controlled by local OS and the other, referred to as *DRALIC* RAM, is controlled by the *DRALIC* driver. The collection of *DRALIC* RAM in all nodes forms a unified system cache for the underlying RAID system. For each node, the RAM, *DRALIC* disk and data disk can be organized as RAPID cache [4].

3. Preliminary Performance Analysis

Firstly, we present a preliminary performance analysis to look at the effects of bus and network delays on the performance potential of the *DRALIC* architecture. The current PCI bus can run at 33-132 MHz with data width of 32 or 64 bits. The memory bandwidth of PCI based system is BW_{mem}=33M*32bits/sec=132MB/sec. A Gigabit Ethernet switch with the transfer speed up to 1Gbps can provide network bandwidth approximately: BW_{net}=100MB/s. The overhead of network operation including both software and hardware is assumed to be OH_{net}=0.2ms[8]. As for disks, we consider a typical SCSI disk drive with specifications as shown in Table 1.

Based on the above disk parameters, we can assume the typical bandwidth of disk to be BW_{dsk}=25MB/s and the overhead of disk to be OH_{dsk}=12ms. The following lists other notations and formulae used in our analysis:

B: data block size (8KB);

N: number of nodes within the *DRALIC* system;

H_{lm}: Local memory hit ratio;

H_{rm}: Remote memory hit ratio;

T_{lm}: Local memory access time (second);

T_{rm}: Remote memory access time (second);

T_{raid}: access time from the distributed RAID (second);

T_{pc}: Average I/O response time of traditional PCs with no cooperative caching (second);

T_{dralic}: Average I/O response time of *DRALIC* system (second);

Table 1: Disk parameters

Model	Capacity	Average Seek Time	Rotational Speed	Average Latency	Transfer rate
UltraStar 18ES	9.1GB	7.0 ms	7200 RPM	4.17ms	187.2-243.7Mbps

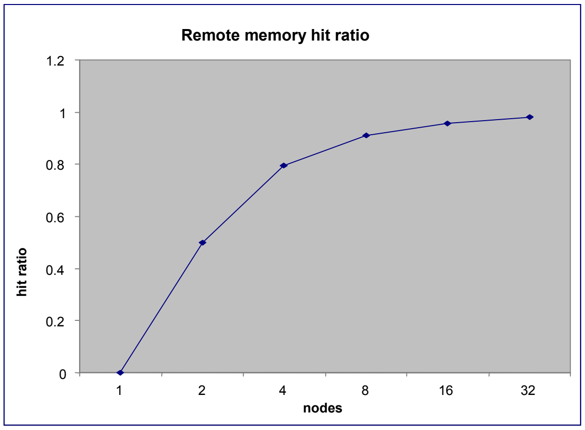
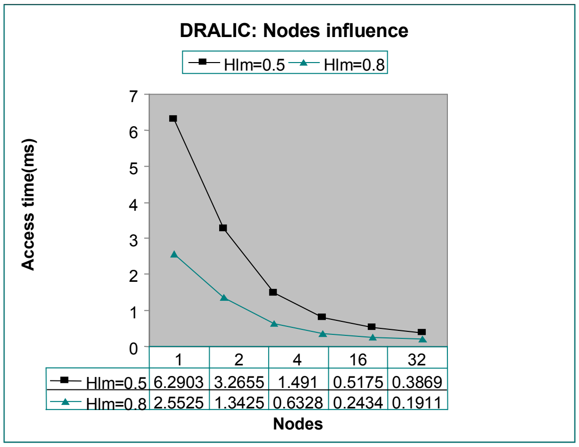


Figure 2: Remote cache miss ratio



	1	2	4	16	32
Hlm=0.5	6.2903	3.2655	1.491	0.5175	0.3869
Hlm=0.8	2.5525	1.3425	0.6328	0.2434	0.1911

Figure 3: Average I/O response time vs. number of nodes

$$T_{lm}=\frac{B}{BW_{mem}}$$

$$T_{rm}=\frac{B}{BW_{net}}+OH_{net}+\frac{B}{BW_{dsk}}$$

$$T_{raid}=\frac{(N-1)B}{N\times BW_{net}}+N\times OH_{net}+\frac{B}{N\times BW_{dsk}}+OH_{dsk}$$

$$T_{pc}=OH_{dsk}+\frac{B}{BW_{dsk}}$$

$$T_{dralic}=H_{lm}\times T_{lm}+(1-H_{lm})\times H_{rm}\times T_{rm}+(1-H_{lm})\times(1-H_{rm})\times T_{raid}$$

With lack of measured hit ratios of remote caches, we assume remote hit ratio to be a logarithm function of number of nodes in the system as shown in Figure 2. It is reasonable to assume that the remote cache hit ratio increases with the number of nodes because more nodes give larger cooperative cache spaces [11]. The exact hit ratio is not significant here since we use the hit ratio as a changing parameter to observe I/O performance as a function of it. From Figure 3, we can see that even with hit ratio of 50%, performance is doubled with two nodes. With remote hit ratio of 80%, a factor of 4 performance improvement can be obtained with 4 nodes. The data in this figure are sufficient to show the potential benefits of *DRALIC*.

4. Simulation Results

To demonstrate the feasibility and performance potential of the proposed *DRALIC*, we designed and implemented a simulator called DralicSim. DralicSim is a program running on every node. In our experiments, 4 nodes running Windows NT are connected through a 100Mbps switch. Four hard drive partitions, one from each node, are combined into a distributed RAID through the DralicSim.

We use the PostMark [6] as our benchmark to measure the results. PostMark measures performance in terms of

Table 2: PostMark Results (Transactions per second)

Tests	Base	2Nodes	3Nodes	4Nodes	Ratio (4Nodes/Base)
Small	78	177	218	243	3.1
Medium	21	53	79	89	4.2
Large	12	26	35	41	3.4

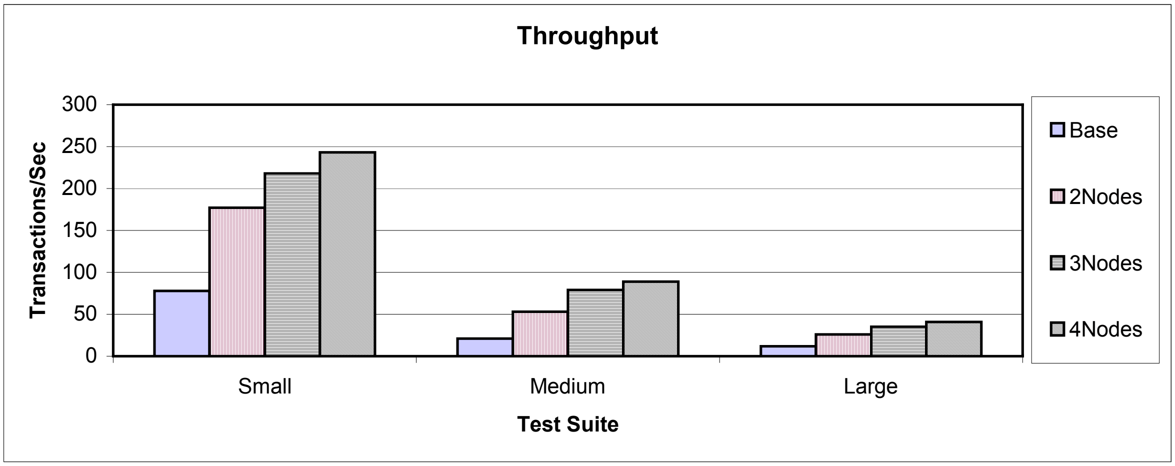


Figure 4: PostMark Results (Transactions per second)

transaction rates in the ephemeral small-file regime by creating a large pool of continually changing files. The file pool is of configurable size. In our tests, PostMark was configured in three different ways as in [6], i.e: 1) small: 1000 intial files and 50000 transactions; 2) medium: 20000 initial files and 50000 transactions; and 3) large: 20000 initial files and 100000 transactions. We left all other PostMark at their default settings.

We configured the DralicSim with 2 nodes (*2Nodes*), 3 nodes (*3Nodes*) and 4 nodes (*4Nodes*) respectively. We tested and compared the results with one node running Windows NT (*Base*). The results of testing are shown in Table 2 and Figure 4, where larger numbers indicate better performance. With 4 nodes connected by DralicSim, the performance gain is up to 4.2, which confirms our preliminary performance analysis above.

5. Conclusions

In this paper we described the architecture and potential performance of *DRALIC*—Distributed RAid and Location Independent Caching, which attempts to combine the disk controller and network controller of existing PCs interconnected by a high-speed LAN switch. Preliminary analysis suggests that a super linear performance improvement can be obtained based on *DRALIC* architecture. Our simulator has the performance gain up to 4.2 with 4 nodes.

Acknowledgements

This research is supported in part by National Science Foundation under Grants CCR-0073377 and MIP9714370. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] T. E. Anderson, M. Dahlin, J. M. Neefe, D. A. Patterson, D. S. Roselli, and R. Y. Wang, "Serverless Network File Systems," In *Proceedings of the Fifteenth ACM Symposium on Operating System Principles,* pp.109-126, Dec. 3-6, 1995

[2] G. A. Gibson and R. V. Meter, "Network Attached Storage Architecture," *Communications of the ACM*, Nov. 2000, Vol.43, No.11

[3] P. Cao, S. B. Lim, S. Venkataraman, and J. Wilkes, "The TickerTAIP Parallel RAID Architecture," *ACM Transactions on Computer Systems* 12(3): 236-269 (1994)

[4] Y. Hu, Q. Yang, and T. Nightingale, "RAPID-Cache: A Reliable and Inexpensive Write Cache for Disk I/O Systems," In *Proceedings of the 5th International Symposium on High Performance Computer Architecture* (HPCA-5), pp.204-213, Orlando, Florida. Jan. 9-13,1999

[5] Intel Developer Forum, *Peer to Peer Computing,* http://developer.intel.com/design/idf/fall2000/presentations/ptp/index.htm, Oct. 2000

[6] J. Katcher, "PostMark: A New File System Benchmark," Technical Report TR3022, Network Appliance

[7] D. E. Long, B. R. Montague, and L. Cabrera, "Swift/RAID: A Distributed RAID System," *Computing Systems* 7(3): 333-359 (1994)

[8] D. A. Patterson and J. L. Hennessy, "Computer Organization & Design: The Hardware/Software Interface," Second Edition, Morgan Kaufmann Publishers, Inc. 1998

[9] C. A. Thekkath, T. Mann, and E. K. Lee, "Frangipani: A Scalable Distributed File System," *16th ACM Symposium on Operating Systems Principles (SOSP-16)*, pp.224-237, Dec. 1997

[10] M. Flouris and E. Markatos, "The Network RamDisk: using remote memory on heterogeneous NOWs," *Cluster Computing: The Journal on Networks, Software, and Applications,* 2(4), pp. 281-293, 1999

[11] G. M. Voelker, E. J. Anderson, T. Kimbrel, M. J. Feeley, J. S. Chase. A. R. Karlin, and H. M. Levy, "Implementing Cooperative Prefetching and Caching in a Globally-Managed Memory System," In *Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems,* pp.33-43, June 22-26,1998