IEEE TRANSACTIONS ON

PARALLEL AND DISTRIBUTED SYSTEMS

A publication of the IEEE Computer Society

MARCH 2002 | VOLUME 13 | NUMBER 3 | ITDSEO | (ISSN 1045-9219)

This resource is also available
on the WWW.
Use MadCat to launch.

IEEE COMPUTER SOCIETY
http://computer.org

IEEE

IEEE COMPUTER SOCIETY

The IEEE Computer Society is an association of people with professional interest in the field of computers. All members of the IEEE are eligible for membership in the Computer Society, as are members of certain professional societies and other computer professionals. Computer Society members will receive this Transactions upon payment of the annual Society membership fee ($35 for IEEE members, $78 for all others) plus an annual subscription fee (paper only: $38; electronic only: $30 combination: $49). For additional membership and subscription information, visit our Web site at http://computer.org/subscribe, send email to help@computer.org, or write to IEEE Computer Society, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314 USA. *Individual subscription copies of Transactions are for personal use only.*

IEEE TRANSACTIONS ON

PARALLEL AND DISTRIBUTED SYSTEMS

IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS is published monthly by the IEEE Computer Society. **IEEE Corporate Office:** Three Park Avenue, 17th Floor, New York, NY 10016-5997 USA. Responsibility for the content rests upon the authors and not upon the IEEE or the IEEE Computer Society. **IEEE Computer Society Publications Office:** 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314 USA. **IEEE Computer Society Headquarters:** 1730 Massachusetts Ave. NW, Washington, DC 20036-1992 USA. **Back issues:** IEEE members $10.00 (first copy only), nonmembers $20.00 per copy. (Note: Add $4.00 postage and handling charge to any order from $1.00 to $50.00, including prepaid orders). Complete price information available on request. Also available in microfiche. **Copyright and Reprint Permissions:** Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee indicated in the code at the bottom of the first page is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923 USA. For all other copying, reprint, or republication permission, write to: Copyrights and Permissions Department, IEEE Publications Administration, 445 Hoes Lane, PO Box 1331, Piscataway, NJ 08855-1331. Periodicals postage paid at New York, NY, and at additional mailing offices. **Postmaster:** Send address changes to IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEM

An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems

John C.S. Lui, *Member, IEEE*, and M.F. Chan, *Student Member, IEEE*

Abstract—Distributed virtual environment (DVE) systems model and simulate the activities of thousands of entities interacting in a virtual world over a wide area network. Possible applications for DVE systems are multiplayer video games, military and industrial trainings, and collaborative engineering. In general, a DVE system is composed of many servers and each server is responsible to manage multiple clients who want to participate in the virtual world. Each server receives updates from different clients (such as the current position and orientation of each client) and then delivers this information to other clients in the virtual world. The server also needs to perform other tasks, such as object collision detection and synchronization control. A large scale DVE system needs to support many clients and this imposes a heavy requirement on networking resources and computational resources. Therefore, how to meet the growing requirement of bandwidth and computational resources is one of the major challenges in designing a scalable and cost-effective DVE system. In this paper, we propose an efficient partitioning algorithm that addresses the scalability issue of designing a large scale DVE system. The main idea is to dynamically divide the virtual world into different partitions and then efficiently assign these partitions to different servers. This way, each server will process approximately the same amount of workload. Another objective of the partitioning algorithm is to reduce the server-to-server communication overhead. The theoretical foundation of our dynamic partitioning algorithm is based on the linear optimization principle. We also illustrate how one can parallelize the proposed partitioning algorithm so that it can efficiently partition a very large scale DVE system. Lastly, experiments are carried out to illustrate the effectiveness of the proposed partitioning algorithm under various settings of the virtual world.

Index Terms—Distributed virtual environment, scalability issue, partitioning algorithm, load balancing, communication reduction, linear optimization.

INTRODUCTION

ADVANCES in multimedia systems, parallel/distributed database systems, and high speed networking technologies enable system designers to build a distributed stem which allows many users to explore and interact der a three dimensional virtual environment. In general, a virtual environment is a virtual world which consists of ny high-resolution 3D graphics sceneries to represent a l-life world. For example, we can have a 3D virtual rld to represent a lecture hall so that hundreds of dents and scientists can listen to a seminar presented Professor Daniel C. Tsui[1] or we can have a large 3D tual world to represent the latest COMDEX show which s many customers reviewing the latest softwares and ctronic gadgets. This type of shared, computer-resident tual world is called a *distributed virtual environment* (DVE) 4]. Like other ground-breaking computer technologies, VE will change the way we learn, work, and interact with her people in the society.

To illustrate how a DVE system can change our lifestyles and the way we handle our business operation, let us consider the following situation. Let's say an architect from New York, a civil and a structural engineer from Paris, a financial planner from Hong Kong, and an interior designer from Tokyo all need to have a business meeting to discuss the designing and financing issues of a new high-rise office complex. Under a DVE setting, these people can convene a meeting in a virtual world without leaving their respective homes and offices. Their meeting can be carried out in a DVE system. These participants can interact with each other in a virtual world of the new high-rise office complex that they are proposing to build. Each participant in this business meeting can virtually *walk around* in the proposed high-rise office building, interact with each other and carry out the discussion. For example, in this virtual high-rise office complex, each participant in the meeting is represented by a 3D object, which is known as an *avatar*. Each participant can walk around in this virtual office building and, in the process, rearrange any 3D object in the environment (e.g., rearrange paintings and furniture or select different kinds of carpet). Any change to a 3D object in this virtual world will be visible to all participants. Participants in this meeting are able to interact with each other in real time, as well as to inquire and to receive any relevant information of the virtual world. For example, participants can query about the credit history of a manufacturer who is responsible to produce the office furniture.

1. A 1998 Noble Prize winner in Physics for the discovery of a new form quantum fluid with fractionally charged excitations.

J.C.S. Lui is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. E-mail: cslui@cse.cuhk.edu.hk.
M.F. Chan is with Poly-Asia Comuter Inc., Kowloon, Hong Kong. E-mail: fei@alumni.cse.cuhk.edu.hk.

nuscript received 11 Jan. 2000; revised 21 Apr. 2001; accepted 13 Aug. 01.
r information on obtaining reprints of this article, please send e-mail to: ls@computer.org, and reference IEEECS Log Number 111201.

RAPID-Cache—A Reliable and Inexpensive Write Cache for High Performance Storage Systems

Yiming Hu, *Senior Member, IEEE*, Tycho Nightingale, and Qing Yang, *Senior Member, IEEE*

Abstract—Modern high performance disk systems make extensive use of nonvolatile RAM (NVRAM) write caches. A single-copy NVRAM cache creates a single point of failure while a dual-copy NVRAM cache is very expensive because of the high cost of NVRAM. This paper presents a new cache architecture called **RAPID-Cache** for *Redundant, Asymmetrically Parallel, and Inexpensive Disk Cache*. A typical RAPID-Cache consists of two redundant write buffers on top of a disk system. One of the buffers is a primary cache made of RAM or NVRAM and the other is a backup cache containing a two-level hierarchy: a small NVRAM buffer on top of a log disk. The small NVRAM buffer combines small write data and writes them into the log disk in large sizes. By exploiting the locality property of I/O accesses and taking advantage of well-known Log-structured File Systems, the backup cache has nearly equivalent write performance as the primary RAM cache. The read performance of the backup cache is not as critical because normal read operations are performed through the primary RAM cache and reads from the backup cache happen only during error recovery periods. The RAPID-Cache presents an asymmetric architecture with a fast-write-fast-read RAM being a primary cache and a fast-write-slow-read NVRAM-disk hierarchy being a backup cache. The asymmetrically parallel architecture and an algorithm that separates actively accessed data from inactive data in the cache virtually eliminate the garbage collection overhead, which are the major problems associated with previous solutions such as Log-structured File Systems and Disk Caching Disk. The asymmetric cache allows cost-effective designs for very large write caches for high-end parallel disk systems that would otherwise have to use dual-copy, costly NVRAM caches. It also makes it possible to implement reliable write caching for low-end disk I/O systems since the RAPID-Cache makes use of inexpensive disks to perform reliable caching. Our analysis and trace-driven simulation results show that the RAPID-Cache has significant reliability/cost advantages over conventional single NVRAM write caches and has great cost advantages over dual-copy NVRAM caches. The RAPID-Cache architecture opens a new dimension for disk system designers to exercise trade-offs among performance, reliability, and cost.

Index Terms—Disks, storage systems, performance, reliability, fault-tolerance.

1 Introduction

Modern disk I/O systems make extensive use of nonvolatile RAM (NVRAM) write caches to facilitate asynchronous write [2], [3], [4], i.e., a write request is acknowledged before the write goes to disk. Such write caches significantly reduce response times of disk I/O systems seen by users, particularly in RAID systems. Large write caches can also improve system throughput by taking advantage of both temporal and spatial localities [2], [5], as data may be overwritten several times or combined together before been written to the disk. IO requests are very bursty [6], requests are often come together with long intervals of relative inactive periods in between. Large write caches also benefit from the the burstiness of write workloads since data coming from bursts can be quickly stored in the cache and written back to the disk later when the system is less busy. Treiber and Menon reported that write caches could reduce disk utilization for writes by an order of magnitude when compared to basic RAID-5 systems [3]. However, the use of write caches introduces two problems: poor reliability and high cost.

Disks are impressively reliable today, with a Mean Time To Failure (MTTF) of up to one million hours. Such a low failure rate, coupled with possible redundancy such as RAID, gives a Mean Time To Data Loss (MTTDL) of several hundreds of millions of hours in a typical RAID-5 system [7]. Adding a single cache in front of a disk system creates a single point of failure, which is vulnerable to data loss. Savage and Wilkes pointed out in [7] that because typical NVRAM technology (battery backed RAM) has a quite low MTTF of 15K hours, a single-copy NVRAM cache suffers significantly higher risk of data loss than results from disk failures. To overcome the reliability problem, some high-end RAID systems use dual-copy caches so that a failure in one cache leaves the other cache intact [2]. When a write request comes, the controller writes two copies of the data independently into the two caches, a primary cache and a backup cache. Besides the reliability problem, NVRAM is also known to be very costly [7], [8], [9] so the size of the

- *Y. Hu is with the Department of Electrical & Computer Engineering and Computer Science, University of Cincinnati, Cincinnati, OH 45221-0030. E-mail: yhu@ececs.uc.edu.*
- *T. Nightingale is with Sun Microsystems, 4150 Network Circle, USAN-208, Santa Clara, CA 95054. E-mail: tycho.nightingale@sun.com.*
- *Q. Yang is with the Department of Electrical and Computer Engineering, University of Rhode Island, Kingston, RI 02881. E-mail: qyang@ele.uri.edu.*

Manuscript received 4 Dec. 2000; accepted 20 Sept. 2001.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 113249.

NVRAM cache is often limited. For example, a major NVRAM manufacturer quoted the price of NVRAM with embedded lithium-cell batteries for $55/MB in quantity as of December 2000. The cost of disks, on the other hand, is about 0.5 cents/MB, which is a difference of four orders of magnitude. Moreover, the cost difference is widening (the difference was three orders of magnitudes two years ago) because prices of disks are falling very rapidly. For a disk system with a reasonably sized write cache, the NVRAM may dominate the cost of the entire system. For example, in a system with 16 disks (40 GB per disk) and an NVRAM write cache of 256 MB, at $55/MB, the NVRAM costs about $14,080, while the total cost of 16 disks is only $3,200 (assuming each 40-GB disk costs $200). If we use dual-copy caches to ease the reliability problem of the single-copy cache, the cost becomes prohibitively high, particularly for large caches. As a result, it is only suitable for the upper echelon of the market [10].

The standard dual-copy write cache system has a *symmetric structure*, where both the primary write cache and the backup write cache have the same size and the same access characteristics—fast read speed and fast write speed. However, the backup cache does not provide any performance benefit to the system during normal operations. Therefore, it is wasteful to use a backup cache identical to the primary cache. What is needed is only a backup cache that can be written to very quickly while its read operations are not as critical since reads from the backup cache occur only during error-recovering periods.

Based on these observations, we propose a new disk cache architecture called *Redundant, Asymmetrically Parallel, Inexpensive Disk Cache*, or **RAPID-Cache** for short, to provide fault-tolerant caching for disk I/O systems inexpensively. The main idea of the RAPID-Cache is to use a conventional, fast-write-fast-read *primary cache* and a nonvolatile, fast-write-slow-read *backup cache*. The primary cache is made of normal NVRAM or DRAM, while the backup cache consists of a small NVRAM cache and a log disk (cache disk). In the backup cache, small and random writes are first buffered in the small NVRAM buffer to form large logs that are written into the *cache disk* later in large transfers, similar to log structured file systems [11], [12], [13], [14]. Because large writes eliminate many expensive small writes, the buffer is quickly made available for additional requests so that the two-level cache appears to the host as a large NVRAM. As a result, the backup cache can achieve the same write speed as the primary cache. The slow-read performance of the backup cache does not affect the system performance since every data block in the backup cache has a copy in the primary cache which can be read at the speed of RAM. The dual cache system here is asymmetric since the primary cache and the backup cache have different sizes and structures. The reliability of the RAPID-Cache is expected to be high since disk is very reliable. The system is also inexpensive because the NVRAM in the backup cache can be very small, ranging from hundreds of KB to several MB and the cost of the disk space is significantly less than that of a large NVRAM. We will show that RAPID-Caches provide much higher reliability compared to single-copy NVRAM caches and much lower cost compared to dual-copy NVRAM caches, without sacrificing performance. On the other hand, because of its low cost, with the *same budget*, RAPID-Caches can have significantly higher performance compared to conventional NVRAM cache architectures by affording much larger primary cache sizes, while still maintaining good reliability.

While the idea of RAPID-Cache can be used in any I/O system, it is particularly suitable for parallel disk systems such as RAID because RAID systems are most likely to be used in environments which require high performance and high reliability. Therefore, we concentrate our study on RAPID-Caches on top of RAID-5 systems in this paper. We have carried out trace-driven simulation experiments as well as analytical studies to evaluate the performance and reliability of the RAPID-Cache. Using real-world traces as well as synthetic traces generated based on realistic workloads [6], [15], we analyze the performance of the RAPID-Cache architecture and compare it with existing disk cache architectures. Numerical results show that the RAPID-Cache has significant performance/cost and reliability advantages over the existing architectures.

The paper is organized as follows: The next section presents the detailed architecture and operations of the RAPID-Cache. Section 3 presents our experimental methodology. Simulation results will be presented in Section 4, followed by an approximate reliability and cost analysis in Section 5. We discuss related work in Section 6 and conclude the paper in Section 7.

2 Architecture and Operations

Fig. 1 shows the basic structure of a RAPID-Cache. It consists of a conventional primary RAM cache and a backup cache. The backup cache is a two-level hierarchy with a small NVRAM on top of a cache disk, similar to DCD [16]. In a RAPID-Cache, every I/O write operation is sent to both the primary cache and the backup cache while read operations are performed using the primary cache only.

For very high overall reliability, the primary cache can be NVRAM to provide redundant protection during a power failure. On the other hand, for low cost systems, the primary cache can be DRAM. During normal operations, the DRAM primary cache and the backup cache contain redundant data. If any one of the two caches fails, data can be reconstructed from the other. During a power failure, data are retained in the backup NVRAM and the cache disk. If both the read cache and the primary write cache are made of DRAM, we can use a unified read/write cache structure, as shown in Fig. 2a, for better cache utilization. A RAPID-Cache with a large unified DRAM primary cache has higher throughput, lower cost, and better reliability than that of a single-copy conventional NVRAM cache. For many applications that require redundant protection during a power failure, a *Triple RAPID-Cache* (shown in Fig. 2b) can be used to build a highly reliable, very large cache system. The idea is to use two low-cost backup caches to support one large primary DRAM cache. During normal operations, the primary cache and the two backup caches provide triple redundancy protection. The two backup caches provide dual-redundancy protection during a power failure. Triple RAPID-Caches are especially suitable for high-end systems

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.