PARALLEL DATA LABORATORY
CARNEGIE MELLON UNIVERSITY

FALL UPDATE

PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • FALL 2014
http://www.pdl.cmu.edu/

PDL CONSORTIUM MEMBERS

Actifio
American Power Conversion
EMC Corporation
Facebook
Google
Hewlett-Packard Labs
Hitachi
Huawei Technologies
Intel Corporation
Microsoft Research
NEC Laboratories
NetApp, Inc.
Oracle Corporation
Samsung Information Systems America
Seagate Technology
Symantec Corporation
Western Digital

CONTENTS

THE PDL PACKET

EDITOR
Joan Digney
CONTACTS
Greg Ganger
PDL Director

Bill Courtright
PDL Executive Director

Karen Lindenfelser
PDL Administrative Manager

The Parallel Data Laboratory

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891
TEL 412-268-6716
FAX 412-268-3010

http://www.pdl.cmu.edu/Publications/

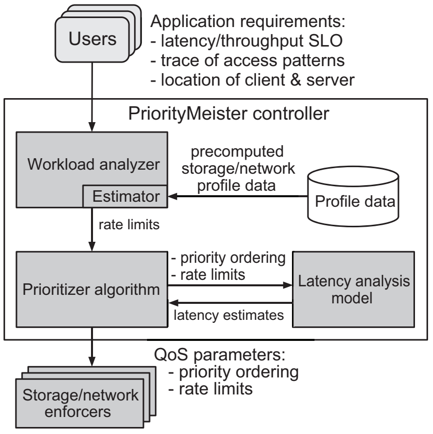
SELECTED RECENT PUBLICATIONS

PriorityMeister: Tail Latency QoS for Shared Networked Storage

Timothy Zhu, Alexey Tumanov, Michael A. Kozuch, Mor Harchol-Balter & Gregory R. Ganger

ACM Symposium on Cloud Computing 2014 (SoCC'14), Seattle, WA, Nov. 2014.

Tail latency service level objectives (SLOs) are an important, but very challenging, problem for cloud computing infrastructures. Existing approaches are effective for average-case performance and low-burstiness workloads, but not for tail latency SLOs under bursty workloads. This paper describes PriorityMeister, a system that combines per-workload priorities and rate limiting to provide tail latency QoS for shared networked storage servicing bursty workloads. PriorityMeister automatically and proactively configures the priorities and rate limits, even for networked storage that involves multiple stages (e.g., shared networks and shared storage servers). In real system experiments and under production trace workloads, PriorityMeister is shown to outperform most recent reactive request scheduling approaches, with more workloads satisfying latency SLOs at higher latency percentiles, while being robust to mis-estimation of underlying storage device performance and containing the effect of misbehaving workloads.



PriorityMeister controller dataflow diagram.

Exploiting Iterative-ness for Parallel ML Computations

Henggang Cui, Alexey Tumanov, Jinliang Wei, Lianghong Xu, Wei Dai, Jesse Haber-Kucharsky, Qirong Ho, Greg R. Ganger, Phil B. Gibbons, Garth A. Gibson & Eric P. Xing

ACM Symposium on Cloud Computing 2014 (SoCC'14), Seattle, WA, Nov 2014.

Many large-scale machine learning (ML) applications use iterative algorithms to converge on parameter values that make the chosen model fit the input data. Often, this approach results in the same sequence of accesses to parameters repeating each iteration. This paper shows that these repeating patterns can and should be exploited to improve the efficiency of the parallel and distributed ML applications that will be a mainstay in cloud computing environments. Focusing on the increasingly popular "parameter server" approach to sharing model parameters among worker threads, we describe and demonstrate

continued on page 6

DISSERTATION ABSTRACT: Trading Freshness for Performance in Distributed Systems

James Cipar

Carnegie Mellon University, SCS

Ph.D. Dissertation
October 3, 2014

Many data management systems are faced with a constant, high-throughput stream of updates. In some cases, these updates are generated externally: a data warehouse system must ingest a stream of external events and update its state. In other cases, they are generated by the application itself: large-scale machine learning frameworks maintain a global shared state, which is used to store the parameters of a statistical model. These parameters are constantly read and updated by the application.

In many cases, there is an trade-off between the freshness of the data returned by read operations and the efficiency of updating and querying the data. For instance, batching many updates together will significantly improve the update throughput for most systems. However, batching introduces a delay between when an update is submitted and when it is available to queries.

In this dissertation, I examine this trade-off in detail. I argue that systems should be designed so that the trade-off can be made by the application, not the data management system. Furthermore, this trade-off should be made at query time, on a per-query basis, not as a global configuration.

To demonstrate this, I describe two novel systems. LazyBase is a data warehouse system designed for metadata management in an archival file storage system. It batches updates and processes them through a pipeline of transformations before applying them to the database, allowing it to achieve very high update throughput. The novel pipeline query mechanism in LazyBase allows applications to select their desired freshness at query time, potentially reading data that is still in the update pipeline and has not yet been applied to the final database.

LazyTables is a distributed machine learning parameter server - a shared storage system for sparse vectors and matrices that make up the bulk of the data in many machine learning applications. To achieve high performance in the face of network delays and performance jitter, it makes extensive use of batching and caching, both in the client and server code. The Stale Synchronous Parallel consistency model, conceived for LazyTables, allows clients to specify how out-of-sync different threads of execution may be.

Andy Pavlo gives an "Overview of Database and NVM Research" at the 2014 PDL Spring Visit Day.

DISSERTATION ABSTRACT: Trading Freshness for Performance in Distributed Systems

Iulian Moraru

Carnegie Mellon University, SCS

Ph.D. Dissertation
July 18, 2014

This thesis describes the design and implementation of state machine replication (SMR) that achieves near-perfect load balancing and availability, near-optimal request processing latency (especially in the wide area), and performance robustness when confronted with failures and slow replicas.

Traditionally, practical replicated state machines have used leader-based implementations of consensus algorithms, because it has been believed that they provide the best performance—highest throughput and lowest latency. At the same time, however, a leader-based approach has many drawbacks: the failure of the leader halts the entire replicated state machine temporarily, the speed of the entire set is determined by the speed of the leader, and, in geo-replicated scenarios, the distance to the leader causes remote clients to experience high latency.

This work shows that leaderless approaches can not only solve these problems and provide the flexibility of a completely decentralized system, but they can also achieve substantially higher performance than leader-based protocols. We introduce a new variant of the Paxos protocol that we call Egalitarian Paxos. In Egalitarian Paxos all replicas perform the same functions simultaneously to ensure better load balancing and availability, lower commit latency and higher performance robustness when compared to previous Paxos variants. We show—both theoret-

PDL students, staff and faculty, along with industry guests gather for the PDL Consortium Speaker Series — a special afternoon of seminars by industry leaders held prior to PDL Spring Visit Day.

continued on page 3

continued from page 2

ically and empirically—that Egalitarian Paxos has the aforementioned benefits when updating the state of a replicated state machine. We then apply the same leaderless design principle to improve the SMR read performance: quorum read leases generalize previously proposed time lease-based approaches to allow arbitrary sets of replicas to perform highly consistent local reads for parts of the replicated state.

THESIS PROPOSAL: Improving System Reliability with Introspective Hardware/Software Fault Monitoring and Prevention for Memory/Storage Devices

Justin Meza, ECE

October 31, 2014

Modern systems rely on the assumption that their hardware components will remain reliable, or free of faults, throughout their operational lifetime. This assumption reduces the burden on the programmer and system designer by alleviating the need to provision for unexpected failures in a system. However, as several recent works have shown, hardware components frequently experience faults during their operational lifetime (a recent study that we performed found that around 1.82% of the machines in a large-scale web services company experienced memory errors at least once per month), motivating the need for systems that can tolerate errors or prevent errors all together. This proposal discusses some of my recent analysis of hardware faults in dynamic random access memory (DRAM) devices at Facebook and outlines my PhD thesis research agenda. Motivated by my initial work, my research plan centers around three main thrusts toward understanding, monitoring, and preventing faults in computing systems focusing on: (1) field study-based statistical fault vector correlation and identification, (2) hardware/software cooperative techniques for proactive fault prevention, and the application of these thrusts to enable (3) introspective hardware/software fault monitoring and reduction.

Peter Klemperer describes his work on "Efficient Hypervisor Based Introspection with Snapshots" to Roger MacNicol (Oracle) at the 2014 PDL Spring Visit Day.

Jin Kyu Kim talks about his research on "STRADS: A Distributed Dynamic Scheduler for Parallel Machine Learning" with Deborah Stokes (EMC) and Jie Yu (Western Digital).

M.S. THESIS: Comparison of Cleaning Performance for SMR Drives

Mukul Singh, INI

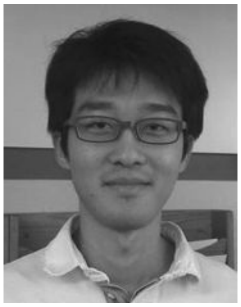
MS Information Networking
May 6, 2014

Shingled Magnetic Recording (SMR) promises to sustain current growth in disk drive capacities with minimal change in the current disk drive technology. Shingling implies overlapping of tracks in a hard drive. Shingling would cause overwrites on down-track sectors with each sector write, hence new interfaces are being proposed to allow host software to exploit SMR with minimal change. An obvious interface is a Shingled Translation Layer which is akin to a Flash Translation Layer. Here the disk can completely hide the layer of remapping and background cleaning, but this comes at the cost of complexity in the disk processor and hard-to-predict performance changes. Other interfaces which enable the host application to handle shingling have been proposed as well. In a strict append model , the disk is divided into fixed sized bands and data is written to a particular band in a strict append order, with cleaning done by resetting the write cursor to the beginning of a band. Another promising interface, Caveat Scriptor, gives the host an address space of all possible sectors. In- order to handle shingling, this interface exposes two drive parameters to determine which sectors may or will not be damaged because of a certain write. These parameters are Drive No Overlap Range (DNOR) and Drive Isolation Distance (DID). This paper will explain these parameters, explain the design of a filesystem designed for this extreme interface, caveat scriptor, and compare the cleaning performance of a filesystem designed for the Caveat Scriptor interface to one designed for the Strict Append interface.

Iulian Moraru and Michael Kaminsky (Intel) discuss Iulian's work on "Egalitarian Paxos" at a 2014 PDL Spring Visit Day poster session.

October 2014

Yoongu Kim Receives Samsung PhD Scholarship



We are pleased to announce that Yoongu Kim, advised by Onur Mutlu, is the inaugural recipient of Samsung USA PhD Fellowship. The fellowship will support Kim for the academic year, and this year; he will be the sole recipient.

July 2014

Pavlo Receives SIGMOD Dissertation Award

Andy Pavlo, assistant professor of computer science, has received the 2014 SIGMOD Jim Gray Doctoral Dissertation Award, which recognizes the best dissertation in the field of databases for the previous year. Pavlo earned his Ph.D. in computer science last year at Brown University. His thesis, "On Scalable Transaction Execution in Partitioned Main Memory Database Management Systems," was based on H-Store, an experimental, distributed main memory database management system. H-Store was the first of a new class of database systems, known as NewSQL, that support highly-concurrent workloads without giving up the transactional guarantees of traditional, relational systems. The system was later commercialized as VoltDB in 2009. The award was presented June 26 at the ACM Special Interest Group on the Management of Data Conference in Snowbird, Utah. He shared this year's prize with Aditya Parameswaran of Stanford University.

--Byron Spice, Carnegie Mellon News, July 2, 2014

June 2014

Hannah Orland Arrives!

Hannah Leah Orland was born June 28, 2014, at 6 lbs 1 oz to Michelle Mazurek and Kyle Orland. All three members of the new family are happy and healthy. Here she is already helping Michelle with her research!

June 2014

PDL INI Group wins Teaching Assistant Award

The Information Networking Institute presents awards to a few graduating students each year in recognition of exemplary work during their time in graduate school.

The Outstanding Student Service Award for Teaching Assistant went to a team of four PDL INI students: Aditya Jaltade, Amod Jaltade, Pratik Shah and Mukul Singh (pictured with Greg Ganger and Rajeev Gandhi). The winners received an engraved award and monetary prize.

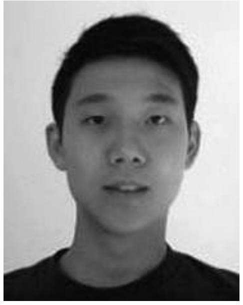
Professors Ganger and Gibson worked with the team of four during the Advanced Storage Systems course. Professor Rajeev Gandhi also nominated Aditya, Amod and Mukul for their assistance with the Fundamentals of Embedded Systems course.

"They gave students a lot of extra help and did it very well by helping them to understand issues or to take a next step forward, without just giving away the solution," said Ganger.

--from INI News at www.ini.cmu.edu/news/; photo © INI@CMU

May 2014

PDL Student Awarded Intel Foundation/SRCEA Graduate Fellowship



ECE doctoral student Kevin Kai-Wei Chang (CMU), who is working with Professor Onur Mutlu on efficient memory systems, has been selected to receive the prestigious Intel Foundation/SRCEA Graduate Fellowship. The fellowship provides tuition and a stipend for up to three years. Kevin recently published a paper at the HPCA 2014 conference on reducing the performance penalty of DRAM refresh, a key limiter of scalability in DRAM memory systems.

May 2014

Mor Harchol-Balter Recipient of Two Teaching Awards

Congratulations to Mor Harchol-Balter (CMU) who received two awards as a result of her teaching (to

continued on page 5

continued from page 4

400 freshmen) of class 21-127 Proof Concepts. The first was at the 2014 CMU Mudge House Dinner with the Deans Honorary Event for Influential Teachers; the second was at 2014 Apple Pie with Alpha Chi Honorary Event for CMU Faculty with Impact on Students.

April 2014
Best Paper Award!

Onur Mutlu (CMU) and co-authors Hyoseung Kim, Dionisio de Niz, Bjorn Andersson, Mark Klein, and Ragunathan (Raj) Rajkumar received the Best Paper Award at the 20th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Berlin, Germany for their work on "Bounding Memory Interference Delay in COTS-based Multi-Core Systems."

April 2014
PDL Paper Presented in Best Paper Session

Samira Khan, an ECE postdoctoral researcher, presented a paper at the 2014 International Symposium on High-Performance Computer Architecture (HPCA), during the Best Paper Session. Dr. Khan was lead author of "Improving Cache Performance by Exploiting Read-Write Disparity." The paper introduces a new mechanism that takes into account the differences in the performance cost of read and write operations in processor caches. It shows that designing a cache that prioritizes cache blocks that serve the more critical read operations can significantly improve system performance.

--ECE News

March 2014
Onur Mutlu Receives Microsoft Prize

Congratulations to Onur Mutlu, who has been awarded the 2014 Microsoft Research Software Engineering Innovation Foundation (SEIF) Award. Microsoft's goals in granting the awards involve continuing to support academic research in software engineering technologies, tools, practices, and teaching methods; stimulating and advancing software engineering practices, development of tools, and programming paradigms; and encouraging the application of software engineering methodologies to data center infrastructure design and management, enabling delivery of cloud-scale services.

The award was granted based on Onur's work on "Improving Datacenter Efficiency and Total Cost of Ownership with Differentiated Software Reliability Analysis and Techniques."

--with info from www.research.microsoft.com

February 2014
PDL Students Receive Bertucci Fellowships

Congratulations to ISTC students Justin Meza and Lavanya Subramanian, who were each awarded a John and Claire Bertucci Fellowship. The

Bertucci Fellowships are awarded to accomplished graduate students who are pursuing doctoral degrees, have passed their PhD qualifying exams and have been admitted to PhD candidacy. The fellowships provide financial support towards their studies and research.

December 2013
Gandhi Awarded SPEC Dissertation Award

Anshul Gandhi (ISTC-CC alum) won the SPEC Dissertation award for his Ph.D. thesis, titled, "Dynamic Server Provisioning for Data Center Power Management," awarded at the International conference on Performance Engineering in March 2014. The Research Group of the Standard Performance Evaluation Corporation (SPEC) selects the annual research prize to be awarded to a Ph.D. student whose thesis is regarded to be an exceptional, innovative contribution in the scope of the SPEC Research Group. Anshul is starting his new position this September as an Assistant Professor at SUNY StonyBrook.

continued from page 1

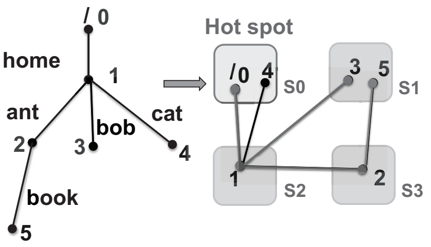
how the repeating patterns can be exploited. Examples include replacing dynamic cache and server structures with static pre-serialized structures, informing prefetch and partitioning decisions, and determining which data should be cached at each thread to avoid both contention and slow accesses to memory banks attached to other sockets. Experiments show that such exploitation reduces per-iteration time by 33–98%, for three real ML workloads, and that these improvements are robust to variation in the patterns over time.

IndexFS: Scaling File System Metadata Performance with Stateless Caching and Bulk Insertion

Kai Ren, Qing Zheng, Swapnil Patil & Garth Gibson

ACM/IEEE Int'l Conf. for High Performance Computing, Networking, Storage and Analysis (SC'14), November 16-21, 2014, New Orleans, LA.

The growing size of modern storage systems is expected to exceed billions of objects, making metadata scalability critical to overall performance. Many existing distributed file systems only focus on providing highly parallel fast access to file data, and lack a scalable metadata service. In this paper, we introduce a middleware design called IndexFS that adds support to existing file systems such as PVFS, Lustre, and HDFS for scalable high-performance operations on metadata and small files. IndexFS uses a table-based architecture that incrementally partitions the namespace on a per-directory basis, preserving server and disk locality for small directories. An optimized log-structured layout is used to store metadata and small files efficiently. We also propose two client-based stormfree caching techniques: bulk namespace insertion for creation intensive workloads such as N-N checkpointing; and stateless consistent metadata caching for hot spot mitigation. By combining these techniques, we have demonstrated IndexFS scaled to 128 metadata servers. Experiments show our out-of-core metadata throughput out-performing existing solutions such as PVFS, Lustre, and HDFS by 50% to two orders of magnitude.



This figure shows how IndexFS distributes a file system directory tree evenly into four metadata servers. Path traversal makes some directories (e.g. root directory) more frequently accessed than others. Thus stateless directory caching is used to mitigate these hot spots.

FIRM: Fair and High-Performance Memory Control for Persistent Memory Systems

Jishen Zhao, Onur Mutlu & Yuan Xie

Proceedings of the 47th International Symposium on Microarchitecture (MICRO), Cambridge, UK, December 2014.

Byte-addressable nonvolatile memories promise a new technology, persistent memory, which incorporates desirable attributes from both traditional main memory (byte-addressability and fast interface) and traditional storage (data persistence). To support data persistence, a persistent memory system requires sophisticated data duplication and ordering control for write requests. As a result, applications that manipulate persistent memory (persistent applications) have very different memory access characteristics than traditional (non-persistent) applications, as shown in this paper. Persistent applications introduce heavy write traffic to contiguous memory regions at a memory channel, which cannot concurrently service read and write requests, leading to memory bandwidth underutilization due to low bank-level parallelism, frequent write queue drains, and frequent bus turnarounds between reads and writes. These characteristics undermine the high-performance and fairness offered by conventional memory scheduling schemes designed for non-persistent applications.

Our goal in this paper is to design a fair and high-performance memory control scheme for a persistent memory based system that runs both persistent and non-persistent applications. Our proposal, FIRM, consists of three key ideas. First, FIRM categorizes request sources as non-intensive, streaming, random and persistent, and forms batches of requests for each source. Second, FIRM strides persistent memory updates across multiple banks, thereby improving bank-level parallelism and hence memory bandwidth utilization of persistent memory accesses. Third, FIRM schedules read and write request batches from different sources in a manner that minimizes bus turnarounds and write queue drains. Our detailed evaluations show that, compared to five previous memory scheduler designs, FIRM provides significantly higher system performance and fairness.

The Blacklisting Memory Scheduler: Achieving High Performance and Fairness at Low Cost

Lavanya Subramanian, Donghyuk Lee, Vivek Seshadri, Harsha Rastogi & Onur Mutlu

Proceedings of the 32nd IEEE International Conference on Computer Design (ICCD), Seoul, South Korea, October 2014.

In a multicore system, applications running on different cores interfere at main memory. This inter-application interference degrades

continued on page 7

continued from page 6

overall system performance and unfairly slows down applications. Prior works have developed application-aware memory request schedulers to tackle this problem. State-of-the-art application-aware memory request schedulers prioritize memory requests of applications that are vulnerable to interference, by ranking individual applications based on their memory access characteristics and enforcing a total rank order.

In this paper, we observe that state-of-the-art application-aware memory schedulers have two major shortcomings. First, ranking applications individually with a total order based on memory access characteristics leads to high hardware cost and complexity. Second, ranking can unfairly slow down applications that are at the bottom of the ranking stack. To overcome these shortcomings, we propose the Blacklisting Memory Scheduler (BLISS), which achieves high system performance and fairness while incurring low hardware cost and complexity. BLISS design is based on two new observations. First, we find that, to mitigate interference, it is sufficient to separate applications into only two groups, one containing applications that cause interference and another containing applications vulnerable to interference, instead of ranking individual applications with a total order. Vulnerable-to-interference group is prioritized over the interference-causing group. Second, we show that this grouping can be efficiently performed by simply counting the number of consecutive requests served from each application – an application that has a large number of consecutive requests served is dynamically classified as interference-causing.

We evaluate BLISS across a wide variety of workloads and system configurations and compare its performance and complexity with five state-of-the-art memory schedulers. Our evaluations show that BLISS achieves 5% better system performance and 25% better fairness than the best-performing previous memory scheduler while greatly reducing critical path latency and hardware area cost of the memory scheduler (by 79% and 43%, respectively).

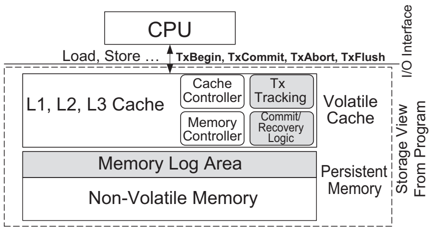
Loose-Ordering Consistency for Persistent Memory

Youyou Lu, Jiwu Shu, Long Sun & Onur Mutlu

Proceedings of the 32nd IEEE International Conference on Computer Design (ICCD), Seoul, South Korea, October 2014.

Emerging non-volatile memory (NVM) technologies enable data persistence at the main memory level at access speeds close to DRAM. In such persistent memories, memory writes need to be performed in strict order to satisfy storage consistency requirements and enable correct recovery from system crashes. Unfortunately, adhering to a strict order for writes to persistent memory significantly degrades system performance as it requires flushing dirty data blocks from CPU caches and waiting for their completion at the main memory in the order specified by the program.

This paper introduces a new mechanism, called Loose-Ordering Consistency (LOC), that satisfies the ordering requirements of persistent memory writes at significantly lower performance degradation than state-of-the-art mechanisms. LOC consists of two key techniques. First, Eager Commit reduces the commit overhead for writes within a transaction by eliminating the need to perform a persistent commit record write at the end of a transaction. We do so by ensuring that we can determine the status of all committed transactions during recovery by storing necessary metadata information statically with blocks of data written to memory. Second, Speculative Persistence relaxes the ordering of writes between transactions by allowing writes to be speculatively written to persistent memory. A speculative write is made visible to software only after its associated transaction commits. To enable this, our mechanism requires the tracking of committed transaction ID and support for multi-versioning in the CPU cache. Our evaluations show that LOC reduces the average performance overhead of strict write ordering from 66.9% to 34.9% on a variety of workloads.



LOC Design Overview.

Using RDMA Efficiently for Key-Value Services

Anuj Kalia, Michael Kaminsky & David G. Andersen

ACM SIGCOMM 2014. Chicago, Illinois, August 17-22, 2014.

This paper describes the design and implementation of HERD, a key-value system designed to make the best use of an RDMA network. Unlike prior RDMA-based key-value systems, HERD focuses its design on reducing network round trips while using efficient RDMA primitives; the result is substantially lower latency, and throughput that saturates modern, commodity RDMA hardware.

HERD has two unconventional decisions: First, it does not use RDMA reads, despite the allure of operations that bypass the remote CPU entirely. Second, it uses a mix of RDMA and messaging verbs, despite the conventional wisdom that the messaging primitives are slow. A HERD client writes its request into the server's memory; the server computes the

continued on page 8

continued from page 7

reply. This design uses a single round trip for all requests and supports up to 19.8 million key-value operations per second with 11 ms average latency. Notably, for small key-value items, our full system throughput is similar to native RDMA read throughput and is over 2X higher than recent RDMA-based key-value systems. We believe that HERD further serves as an effective template for the construction of RDMA-based datacenter services.

CHIPS: Content-based Heuristics for Improving Photo Privacy for Smartphones

Jiaqi Tan, Utsav Drolia, Rolando Martins, Rajeev Gandhi & Priya Narasimhan

7th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec), July 2014.

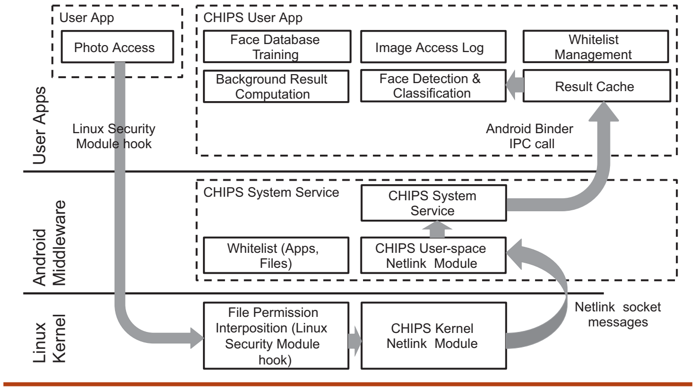
The Android permissions system provides all-or-nothing access to users' photos stored on smartphones, and the permissions which control access to stored photos can be confusing to the average user. Our analysis found that 73% of the top 250 free apps on the Google Play store have permissions that may not reflect their ability to access stored photos. We pro- pose CHIPS, a unique content-based fine-grained runtime access control system for stored photos for Android which requires minimal user assistance, runs entirely locally, and provides low-level enforcement. CHIPS can recognize faces with minimal user training to deny apps access to photos with known faces. CHIPS's privacy identification has low overheads as privacy checks are cached, and is accurate, with false-positive and false-negative rates of less than 8%.

Towards Secure Execution of Untrusted Code for Mobile Edge-Clouds

Jiaqi Tan, Utsav Drolia, Rajeev Gandhi & Priya Narasimhan

Poster at 7th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec), July 2014.

Mobile edge-clouds have nodes comprised entirely of mobile devices, and they enable applications across these devices without central coordination, pooling both compute/storage resources and data on these devices. While mobile edge-clouds enable applications to run even when there is poor or no infrastructure connectivity (e.g. packed stadiums, disaster response) they can pose serious risks to the security of mobile devices when nodes execute arbitrary code from untrusted clients. We propose a system to enable mobile edge-cloud nodes to securely execute code submitted by untrusted clients, by automatically proving that machine code programs meet desired memory and control-flow safety properties before programs are executed. Our system will enable mobile edge-cloud nodes to be versatile, as they can safely run arbitrary code submitted by clients with the assurance that desired safety properties are met, without needing code vetting or establishing of client identities ahead of time.



Architecture of CHIPS.

Will They Blend?: Exploring Big Data Computation atop Traditional HPC NAS Storage

Ellis H. Wilson III, Mahmut T. Kandemir & Garth Gibson

The 34th International Conference on Distributed Computing Systems, ICDCS 2014, June 30 - July 3, 2014, Madrid, Spain.

The Apache Hadoop framework has rung in a new era in how data-rich organizations can process, store, and analyze large amounts of data. This has resulted in increased potential for an infrastructure exodus from the traditional solution of commercial database ad-hoc analytics on network-attached storage (NAS). While many data-rich organizations can afford to either move entirely to Hadoop for their Big Data analytics, or to maintain their existing traditional infrastructures and acquire a new set of infrastructure solely for Hadoop jobs, most supercomputing centers do not enjoy either of those possibilities. Too much of the existing scientific code is tailored to work on massively parallel file systems unlike the Hadoop Distributed File System (HDFS), and their datasets are too large to reasonably maintain and/or ferry between two distinct storage systems. Nevertheless, as scientists search for easier-to-program frameworks with a lower time-to-science to postprocess their huge datasets after execution, there is increasing pressure to enable use of MapReduce within these traditional High Performance Computing (HPC) architectures.

Therefore, in this work we explore potential means to enable use of the easy-to-program Hadoop MapReduce framework without requiring a complete infrastructure overhaul from existing HPC NAS solutions. We demonstrate that retaining function-dedicated resources like NAS is not only possible, but can even be effected efficiently with MapReduce. In our

continued on page 9

continued from page 8

exploration, we unearth subtle pitfalls resultant from this mashup of new-era Big Data computation on conventional HPC storage and share the clever architectural configurations that allow us to avoid them. Last, we design and present a novel Hadoop File System, the Reliable Array of Independent NAS File System (RainFS), and experimentally demonstrate its improvements in performance and reliability over the previous architectures we have investigated.

Exploiting Bounded Staleness to Speed up Big Data Analytics

Henggang Cui, James Cipar, Qirong Ho, Jin Kyu Kim, Seunghak Lee, Abhimanu Kumar Jinliang Wei, Wei Dai, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson & Eric P. Xing

2014 USENIX Annual Technical Conference (ATC'14). June 19-20, 2014. Philadelphia, PA.

Many modern machine learning (ML) algorithms are iterative, converging on a final solution via many iterations over the input data. This paper explores approaches to exploiting these algorithms' convergent nature to improve performance, by allowing parallel and distributed threads to use loose consistency models for shared algorithm state. Specifically, we focus on bounded staleness, in which each thread can see a view of the current intermediate solution that may be a limited number of iterations out-of-date. Allowing staleness reduces communication costs (batched updates and cached reads) and synchronization (less waiting for locks or straggling threads). One approach is to increase the number of iterations between barriers in the oft-used Bulk Synchronous Parallel (BSP) model of parallelizing, which mitigates these costs when all threads proceed at the same speed. A more flexible approach, called Stale Synchronous Parallel (SSP), avoids barriers and allows threads to be a bounded number of iterations ahead of the current slowest thread. Extensive experiments with ML algorithms for topic modeling, collaborative filtering, and PageRank show that both approaches significantly increase convergence speeds, behaving similarly when there are no stragglers, but SSP outperforms BSP in the presence of stragglers.

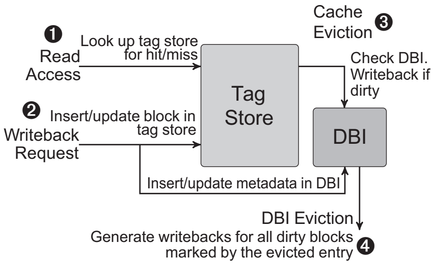
The Dirty-Block Index

Vivek Seshadri, Abhishek Bhowmick, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch & Todd C. Mowry

41st International Symposium on Computer Architecture, June, 2014.

On-chip caches maintain multiple pieces of metadata about each cached block—e.g., dirty bit, coherence information, ECC. Traditionally, such metadata for each block is stored in the corresponding tag entry in the tag store. While this approach is simple to implement and scalable, it necessitates a full tag store lookup for any metadata query—resulting in high latency and energy consumption. We find that this approach is inefficient and inhibits several cache optimizations.

In this work, we propose a new way of organizing the dirty bit information that enables simpler and more efficient implementation of several optimizations. In our proposed approach, we remove the dirty bits from the tag store and organize it differently in a structure, which we call the Dirty-Block Index (DBI). The organization of DBI is simple: it consists of multiple entries, each corresponding to some row in DRAM. A bit vector in each entry tracks whether each block in the corresponding DRAM row is dirty or not. We demonstrate the effectiveness of DBI by using it to simultaneously implement three optimizations proposed by prior work: 1) Aggressive DRAM-aware writeback, 2) Bypassing cache lookups, and 3) Heterogenous ECC for clean/dirty blocks. DBI, with all three optimization enabled, improves performance by 31% compared to baseline (6% compared to the best previous mechanism) while reducing overall area cost by 8% compared to prior approaches.


❶
Read Access
Look up tag store for hit/miss
❷
Writeback Request
Insert/update block in tag store
Tag Store
Cache Eviction ❸
Check DBI. Writeback if dirty
DBI
Insert/update metadata in DBI
DBI Eviction
Generate writebacks for all dirty blocks marked by the evicted entry ❹

Operation of a cache with DBI.

Characterizing Application Memory Error Vulnerability to Optimize Datacenter Cost via Heterogeneous-Reliability Memory

Yixin Luo, Sriram Govindan, Bikash Sharma, Mark Santaniello, Justin Meza, Aman Kansal, Jie Liu, Badriddine Khessib, Kushagra Vaid & Onur Mutlu

Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Atlanta, GA, June 2014.

Memory devices represent a key component of datacenter total cost of ownership (TCO), and techniques used to reduce errors that occur on these devices increase this cost. Existing approaches to providing reliability for memory devices pessimistically treat all data is equally vulnerable to memory errors. Our key insight is that there exists a diverse spectrum of tolerance to memory errors in new data-intensive applications, and that traditional one-size-fits-all memory reliability techniques are inefficient in terms of cost. For example, we found that while traditional error protection increases memory system cost by 12.5%, some applications can achieve 99.00% availability on a single server with a large number of memory errors

continued on page 10

continued from page 9

without any error protection. This presents an opportunity to greatly reduce server hardware cost by provisioning the right amount of memory reliability for different applications.

Toward this end, in this paper, we make three main contributions to enable highly-reliable servers at low datacenter cost. First, we develop a new methodology to quantify the tolerance of applications to memory errors. Second, using our methodology, we perform a case study of three new data intensive workloads (an interactive web search application, an in-memory key–value store, and a graph mining framework) to identify new insights into the nature of application memory error vulnerability. Third, based on our insights, we propose several new hardware/software heterogeneous-reliability memory system designs to lower datacenter cost while achieving high reliability and discuss their tradeoffs. We show that our new techniques can reduce server hardware cost by 4.7% while achieving 99.90% single server availability.

Exact Analysis of the M/M/k/setup Class of Markov Chains via Recursive Renewal Reward

Anshul Gandhi, Sherwin Doroudi, Mor Harchol-Balter & Alan Scheller-Wolf

Queueing Systems: Theory and Applications vol. 77, no. 2, 2014, pp. 177-209. June 2014.

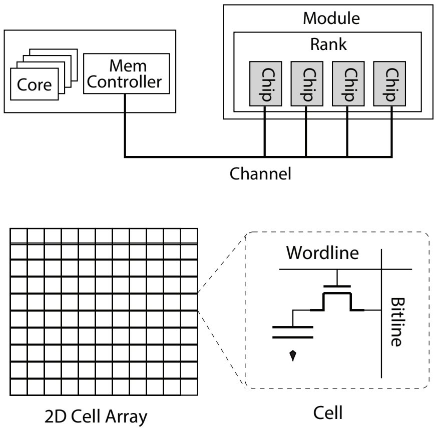
The M/M/k/setup model, where there is a penalty for turning servers on, is common in data centers, call centers, and manufacturing systems. Setup costs take the form of a time delay, and sometimes there is additionally a power penalty, as in the case of data centers. While the M/M/1/setup was exactly analyzed in 1964, no exact analysis exists to date for the M/M/k/setup with k>1. In this paper, we provide the first exact, closed-form analysis for the M/M/k/setup and some of its important variants including systems in which idle servers delay for a period of time before turning off or can be put to sleep. Our analysis is made possible by a new way of combining renewal reward theory and recursive techniques to solve Markov chains with a repeating structure. Our renewal-based approach uses ideas from renewal reward theory and busy period analysis to obtain closed-form expressions for metrics of interest such as the transform of time in system and the transform of power consumed by the system. The simplicity, intuitiveness, and versatility of our renewal-based approach makes it useful for analyzing Markov chains far beyond the M/M/k/setup. In general, our renewal-based approach should be used to reduce the analysis of any 2-dimensional Markov chain which is infinite in at most one dimension and repeating to the problem of solving a system of polynomial equations. In the case where all transitions in the repeating portion of the Markov chain are skip-free and all up/down arrows are unidirectional, the resulting system of equations will yield a closed-form solution.

The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study

Samira Khan, Donghyuk Lee, Yoongu Kim, Alaa Alameldeen, Chris Wilkerson & Onur Mutlu

Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'14), June 2014.

As DRAM cells continue to shrink, they become more susceptible to retention failures. DRAM cells that permanently exhibit short retention times are fairly easy to identify and repair through the use of memory tests and row and column redundancy. However, the retention time of many cells may vary over time due to a property called Variable Retention Time (VRT). Since these cells intermittently transition between failing and non-failing states, they are particularly difficult to identify through memory tests alone. In addition, the high temperature packaging process may aggravate this problem as the susceptibility of cells to VRT increases after the assembly of DRAM chips. A promising alternative to manufacture-time testing is to detect and mitigate retention failures after the system has become operational. Such a system would require mechanisms to detect and mitigate retention failures in the field, but would be responsive to retention failures introduced after system assembly and could dramatically reduce the cost of testing, enabling much longer tests than are practical with manufacturer testing equipment.



DRAM organization (above), and DRAM cells in a rank (below).

In this paper, we analyze the efficacy of three common error mitigation techniques (memory tests, guardbands, and error correcting codes (ECC)) in real DRAM chips exhibiting both intermittent and permanent retention failures. Our analysis allows us to quantify the efficacy of recent system-level error mitigation mechanisms that build upon these techniques. We revisit prior works in the context of the experimental data we present, showing that our measured results significantly impact these works' conclusions. We

continued on page 11

continued from page 10

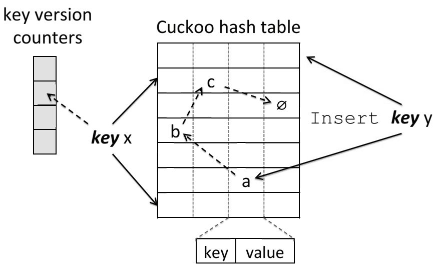
find that mitigation techniques that rely on run-time testing alone [38, 27, 50, 26] are unable to ensure reliable operation even after many months of testing. Techniques that incorporate ECC [4, 52], however, can ensure reliable DRAM operation after only a few hours of testing. For example, VS-ECC [4], which couples testing with variable strength codes to allocate the strongest codes to the most error-prone memory regions, can ensure reliable operation for 10 years after only 19 minutes of testing. We conclude that the viability of these mitigation techniques depend on efficient online profiling of DRAM performed without disrupting system operation.

Algorithmic Improvements for Fast Concurrent Cuckoo Hashing

Xiaozhou Li, David G. Andersen, Michael Kaminsky & Michael J. Freedman

Proceedings of the European Conference on Computer Systems (EuroSys '14), April 2014.

Fast concurrent hash tables are an increasingly important building block as we scale systems to greater numbers of cores and threads. This paper presents the design, implementation, and evaluation of a high-throughput and memory-efficient concurrent hash table that supports multiple readers and writers. The design arises from careful attention to systems-level optimizations such as minimizing critical section length and reducing interprocessor coherence traffic through algorithm re-engineering. As part of the architectural basis for this engineering, we include a discussion of our experience and results adopting Intel's recent hardware transactional memory (HTM) support to this critical building block. We find that naively allowing concurrent access using a coarse-grained lock on existing data structures reduces overall performance with more threads. While HTM mitigates this slowdown somewhat, it does not eliminate it. Algorithmic optimizations that benefit both HTM and designs for fine-grained locking are needed to achieve high performance.



Cuckoo hash table overview: Each key is mapped to 2 buckets by hash functions and associated with 1 version counter. ø represents an empty slot. "a→b→c→ø" is a cuckoo path to make one bucket available to insert key *y*.

Our performance results demonstrate that our new hash table design—based around optimistic cuckoo hashing—outperforms other optimized concurrent hash tables by up to 2.5x for write-heavy workloads, even while using substantially less memory for small key-value items. On a 16-core machine, our hash table executes almost 40 million insert and more than 70 million lookup operations per second.

Towards Wearable Cognitive Assistance

Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai & Mahadev Satyanarayanan

Proceedings of the 12th ACM International Conference on Mobile Computing, Systems and Services (MobiSys'14), June 2014.

We describe the architecture and prototype implementation of an assistive system based on Google Glass devices for users in cognitive decline. It combines the first-person image capture and sensing capabilities of Glass with remote processing to perform real-time scene interpretation. The system architecture is multi-tiered. It offers tight end-to-end latency bounds on compute-intensive operations, while addressing concerns such as limited battery capacity and limited processing capability of wearable devices. The system gracefully degrades services in the face of network failures and unavailability of distant architectural tiers.

Bounding Memory Interference Delay in COTS-based Multi-Core Systems

Hyoseung Kim, Dionisio de Niz, Björn Andersson, Mark Klein, Onur Mutlu & Ragunathan (Raj) Rajkumar

Proceedings of the 20th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Berlin, Germany, April 2014.

In commercial-off-the-shelf (COTS) multi-core systems, a task running on one core can be delayed by other tasks running simultaneously on other cores due to interference in the shared DRAM main memory. Such memory interference delay can be large and highly variable, thereby posing a significant challenge for the design of predictable real-time systems. In this paper, we present techniques to provide a tight upper bound on the worst-case memory interference in a COTS-based multi-core system. We explicitly model the major resources in the DRAM system, including banks, buses and the memory controller. By considering their timing characteristics, we analyze the worst-case memory interference delay imposed on a task by other tasks running in parallel. To the best of our knowledge, this is the first work bounding the request re-ordering effect of COTS memory controllers. Our work also enables the quantification of the extent by which memory interference can be reduced by partitioning DRAM banks. We evaluate our approach on a commodity multi-core platform running Linux/RK. Experimental results show that our approach provides an upper

continued on page 12

continued from page 11

bound very close to our measured worst-case interference.

Improving Cache Performance by Exploiting Read-Write Disparity

Samira Khan, Alaa R. Alameldeen, Chris Wilkerson, Onur Mutlu, Daniel A. Jiménez

Proceedings of the 20th International Symposium on High-Performance Computer Architecture (HPCA), Orlando, FL, February 2014. Best paper session.

Cache read misses stall the processor if there are no independent instructions to execute. In contrast, most cache write misses are off the critical path of execution, since writes can be buffered in the cache or the store buffer. With few exceptions, cache lines that serve loads are more critical for performance than cache lines that serve only stores. Unfortunately, traditional cache management mechanisms do not take into account this disparity between read-write criticality. The key contribution of this paper is the new idea of distinguishing between lines that are reused by reads versus those that are reused only by writes to focus cache management policies on the more critical read lines. We propose a Read-Write Partitioning (RWP) policy that minimizes read misses by dynamically partitioning the cache into clean and dirty partitions, where partitions grow in size if they are more likely to receive future read requests. We show that exploiting the differences in read-write criticality provides better performance over prior cache management mechanisms. For a single-core system, RWP provides 5% average speedup across the entire SPEC CPU2006 suite, and 14% average speedup for cache-sensitive benchmarks, over the baseline LRU replacement policy. We also show that RWP can perform within 3% of a new yet complex instruction-address-based technique, Read Reference Predictor (RRP), that bypasses cache lines which are unlikely to receive any read requests, while requiring only 5:4% of RRP's state overhead. On a 4-core system, our RWP mechanism improves system throughput by 6% over the baseline and outperforms three other state-of-the-art mechanisms we evaluate.

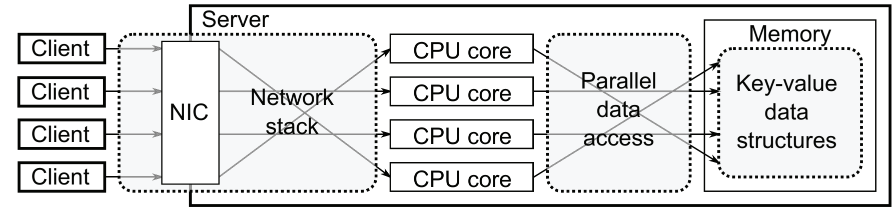
MICA: A Holistic Approach to Near-Line-Rate In-Memory Key-Value Caching on General-Purpose Hardware

Hyeontaek Lim, Dongsu Han, David G. Andersen & Michael Kaminsky

11th USENIX Symposium on Networked Systems Design and Implementation (NSDI'14), April 2014.

MICA is a scalable in-memory key-value store that handles 65.6 to 76.9 million key-value operations per second using a single general-purpose multi-core system. MICA is over 4–13.5x faster than current state-of-the-art systems, while providing consistently high throughput over a variety of mixed read and write workloads.

MICA takes a holistic approach that encompasses all aspects of request handling, including parallel data access, network request handling, and data structure design, but makes unconventional choices in each of the three domains. First, MICA optimizes for multi-core architectures by enabling parallel access to partitioned data. Second, for efficient parallel data access, MICA maps client requests directly to specific CPU cores at the server NIC level by using client-supplied information and adopts a light-weight networking stack that bypasses the kernel. Finally, MICA's new data structures—circular logs, lossy concurrent hash indexes, and bulk chaining—handle both read- and write-intensive workloads at low overhead.



Components of in-memory key-value stores. MICA's key design choices.

Scalable, High Performance Ethernet Forwarding with CuckooSwitch

Dong Zhou, Bin Fan, Hyeontaek Lim, David G. Andersen & Michael Kaminsky

Proceedings 9th International Conference on emerging Networking EXperiments and Technologies (CoNEXT), Dec. 2013.

Several emerging network trends and new architectural ideas are placing increasing demand on forwarding table sizes. From massive-scale datacenter networks running millions of virtual machines to flow-based software-defined networking, many intriguing design options require FIBs that can scale well beyond the thousands or tens of thousands possible using today's commodity switching chips.

This paper presents CUCKOOSWITCH, a software-based Ethernet switch design built around a memory-efficient, high-performance, and highly-concurrent hash table for compact and fast FIB lookup. We show that CUCKOOSWITCH can process 92.22 million minimum-sized packets per second on a commodity server equipped with eight 10 Gbps Ethernet interfaces while maintaining a forwarding table of one billion forwarding entries. This rate is the maximum packets per second achievable across the underlying hardware's PCI buses.