PARALLEL DATA LABORATORY
CARNEGIE MELLON UNIVERSITY

PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • SPRING 2015
http://www.pdl.cmu.edu/

AN INFORMAL PUBLICATION FROM ACADEMIA'S PREMIERE STORAGE SYSTEMS RESEARCH CENTER DEVOTED TO ADVANCING THE STATE OF THE ART IN STORAGE AND INFORMATION INFRASTRUCTURES.

CONTENTS

DBMSs for NVM 1
Director's Letter 2
Year in Review 4
Recent Publications 5
PDL News & Awards 8
Dissertations & Proposals 14

PDL CONSORTIUM MEMBERS

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

Database Management Systems for Non-Volatile Memory

Joy Arulraj, Andy Pavlo & Joan Digney

Changes in computer trends have given rise to new on-line transaction processing (OLTP) applications that support a large number of concurrent users and systems. What makes these modern applications unlike their predecessors is the scale at which they ingest information. Database management systems (DBMSs) are the critical component of these applications because they are responsible for ensuring transaction operations execute in the correct order and that changes are not lost after a crash. Optimizing a DBMS' performance is important because it determines how quickly an application can take in new information and how quickly the information can be used to make new decisions.

DBMSs have always dealt with the trade-off between volatile and non-volatile storage devices. In order to retain data after a loss of power, the DBMS must write that data to a non-volatile device, such as a SSD or HDD, but such devices only support slow, bulk data transfers as blocks. Contrast this with volatile DRAM, where a DBMS can quickly read and write a single byte from these devices, but all data is lost if power is lost.

There are inherent physical limitations that prevent DRAM from scaling to capacities beyond today's levels. Using a large amount of DRAM consumes a lot of energy—up to 40% of the overall power consumed by a server—since it requires periodic refreshing to preserve data even if it is not actively used.

Although flash-based SSDs have better storage capacities and use less energy than DRAM, they have other issues that make them less than ideal. For example, they are much slower than DRAM and only support unwieldy block-based access methods. This means that if a transaction updates a single byte of data stored on an SSD, then the DBMS must write the change out as a block (typically 4 KB). This is problematic and inefficient for OLTP applications that make many

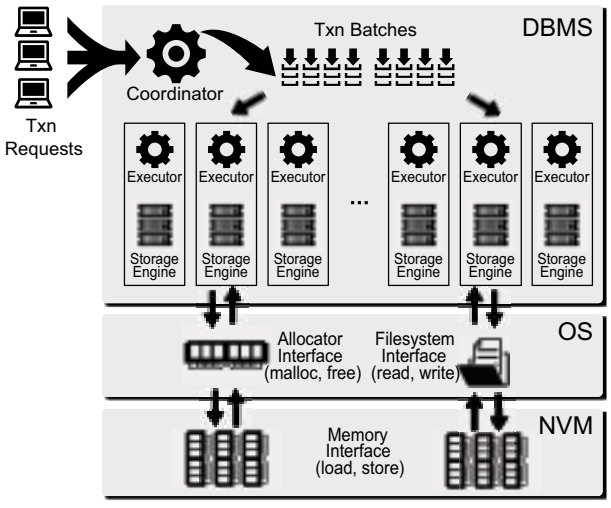


Figure 1: An overview of the architecture of the DBMS testbed.

continued on page 11

FROM THE DIRECTOR'S CHAIR

Greg Ganger

Hello from fabulous Pittsburgh!

It has been another great year for the Parallel Data Lab. Some highlights include rapid growth in database systems research, more great activity in cloud computing and "Big Data" systems, record-breaking demand for our storage systems and cloud classes, and great recognition of PDL faculty, students, and staff. Along the way, many students graduated and joined PDL Consortium companies, new students joined the PDL, and many cool papers have been published. Let me highlight a few things.

We continue to be energized by finding ourselves at the core of two of today's largest growth areas, in addition to new storage systems generally: cloud computing and Big Data. While we didn't coin either term, PDL has been active in both areas for a long time, starting before the buzzwords arose. We continue to explore systems approaches for supporting large-scale machine learning (a primary component of Big Data analytics), expand Masters program activities in both areas, and lead the 6-institution Intel Science and Technology Center for Cloud Computing (ISTC-CC).

On the education front, our efforts to provide Masters students with excellent foundations in storage systems, cloud technologies, and Big Data systems have almost been too successful. OK, not really, but demand for the storage system and cloud classes that we have created has exploded. This Spring, the storage systems class that Garth and I have taught for over 10 years had 100 students enrolled and another 100 on the wait list. So, we'll be offering it again in the Fall. In addition to our lectures, the class featured five corporate guest lecturers (thank you, PDL Consortium members!) bringing insight on real-world storage, trends, and futures. The storage class and the cloud class serve several Masters programs, including the Masters program on data science systems that Garth has developed. That latter trains students with strong practical skills in the creation and exploitation of systems for Big Data analytics, including allowing extended internships to satisfy the program's capstone project requirement—something which several PDL companies have explored.

As noted last year, and highlighted by the front page article, database systems research is back to PDL in a big way. Andy has brought great energy, launching new projects and collaborating with many other PDL faculty to quickly re-establish a broad, strong database activity. Look for more great things on that front coming soon, including work on automated database tuning, deduplication in databases, incremental computation, and more exploitation of NVM in databases.

We continue to find great opportunities and impact from working closely with Carnegie Mellon's excellent machine learning faculty to explore better system support for Big Data analytics. Early approaches like Map-Reduce are good for very simple data processing, advanced machine learning requires different approaches to achieve its potential. PDL has long found great success in such cross-domain collaboration, and it continues to result in development of powerful building blocks for future data science in practice. Indeed, it is clear that no single system architecture is going to serve the breadth of data analytics styles and activities—multiple approaches will have a role, each specialized for different classes of tasks.

The breadth of analytics frameworks and other cloud computing activities leads to challenging resource scheduling challenges. For example, our Tetrisched project is developing new ways of allowing users to express their per-job resource type

THE PDL PACKET

The Parallel Data Laboratory
School of Computer Science
Department of ECE
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891
VOICE 412•268•6716
FAX 412•268•3010

PUBLISHER
Greg Ganger

EDITOR
Joan Digney

The PDL Packet is published once per year to update members of the PDL Consortium. A pdf version resides in the Publications section of the PDL Web pages and may be freely distributed. Contributions are welcome.

THE PDL LOGO

Skibo Castle and the lands that comprise its estate are located in the Kyle of Sutherland in the northeastern part of Scotland. Both 'Skibo' and 'Sutherland' are names whose roots are from Old Norse, the language spoken by the Vikings who began washing ashore regularly in the late ninth century. The word 'Skibo' fascinates etymologists, who are unable to agree on its original meaning. All agree that 'bo' is the Old Norse for 'land' or 'place,' but they argue whether 'ski' means 'ships' or 'peace' or 'fairy hill.'

Although the earliest version of Skibo seems to be lost in the mists of time, it was most likely some kind of fortified building erected by the Norsemen. The present-day castle was built by a bishop of the Roman Catholic Church. Andrew Carnegie, after making his fortune, bought it in 1898 to serve as his summer home. In 1980, his daughter, Margaret, donated Skibo to a trust that later sold the estate. It is presently being run as a luxury hotel.

FACULTY

Greg Ganger (pdl director)
412•268•1297
ganger@ece.cmu.edu

David Andersen
Lujo Bauer
Chuck Cranor
Lorrie Cranor
Christos Faloutsos
Eugene Fink
Rajeev Gandhi
Garth Gibson
Seth Copen Goldstein
Mor Harchol-Balter
Todd Mowry
Onur Mutlu
Priya Narasimhan
David O'Hallaron
Andy Pavlo
Majd Sakr
M. Satyanarayanan
Srinivasan Seshan
Alex Smola
Hui Zhang

STAFF MEMBERS

Bill Courtright, 412•268•5485
(pdl executive director) wcourtright@cmu.edu
Karen Lindenfelser, 412•268•6716
(pdl administrative manager) karen@ece.cmu.edu
Joan Digney
Chad Dougherty
Zisimos Economou
Mitch Franzos
Otgonpurev Mendsaikhan
Charlene Zang

VISITING RESEARCHERS / POST DOCS

Saugata Ghose
Samira Khan
Rolando Martins
Jun Nemoto
Raja Sambasivan
Sadahiro Sugimoto

GRADUATE STUDENTS

Joy Arulraj
Rachata Ausavarungnirun
Ravi Chandra Bandlamudi
Vinaykumar Bhat
Ben Blum
Amirali Boroumand
Fiona Britto
Lei Cao
Kevin Chang
Henggang Cui
Utsav Drolia
Jian Fang
Kristen Scholes Gardner
Omkar Gawde
Aaron Harlap
Kevin Hsieh
Gaurav Jain
Junchen Jiang
Wesley Jin
Saurabh Arun Kadekodi
Anuj Kalia
Mike Kasick
Jin Kyu Kim
Elie Krevat
Conglong Li
Mu Li
Yang Li
Hyeontaek Lim
Yixin Luo
Thomas Marshall
Justin Meza
Nathan Mickulicz
Jun Woo Park
Swapnil Pimpale
Kai Ren
Wolfgang Richter
Rohan Sehgal
Vivek Seshadri
Lavanya Subramanian
Jiaqi Tan
Alexey Tumanov
Dana Van Aken
Nandita Vijaykumar
Hui Wang
Jinliang Wei
Lin Xiao
Hongyi Xin
Lianghong Xu
HanBin Yoon
Huanchen Zhang
Rui Zhang
Qing Zheng
Dong Zhou
Timothy Zhu

FROM THE DIRECTOR'S CHAIR

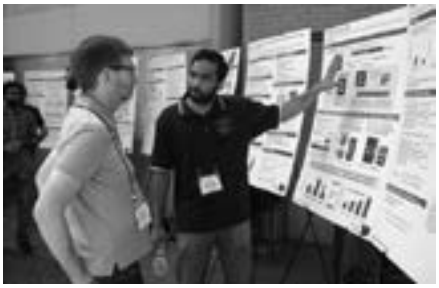
preferences (e.g., machine locality or hardware accelerators) and then exploring the trade-offs among them to maximize utility of the public and/or private cloud infrastructure. As another example, we are also exploring new approaches to providing for end-to-end latency SLOs, especially tail latency, in shared network storage.

Naturally, PDL's long-standing focus on scalable storage continues strongly. As always, a primary challenge is metadata scaling, and PDL researchers are exploring several approaches to dealing with scale along different dimensions. In fact, the paper describing one new approach, called IndexFS, was named Best Paper at Supercomputing 2014. IndexFS is table-based middleware that combines efficient metadata storage with support for bulk namespace and statement consistent caching to provide unprecedented metadata scalability.

We continue to explore ways of exploiting the exciting new underlying storage technologies, such as NVM and Flash SSDs, to improve systems. One example is the front-page article, along with hybrid memory architectures, new key-value and index architectures, and other work. On the other end of the storage hierarchy, shingled magnetic recording (SMR) is changing the way the disk works, and our exploration of new interface styles continues, ranging from hiding SMR-ness behind an FTL-like layer to fully exposing it and leaving it to the host to manage—we call that "caveat scriptor."

Many other ongoing PDL projects are also producing cool results. Our work on elastic storage, such as SpringFS, has paved the way for rapid adaptive change to the set of nodes providing storage performance. We also continue to explore new approaches to cloud offload and to inspecting the storage state of cloud VMs without modifying the software running in those VMs, enabling improved performance, manageability, and security. Our continued operation of private clouds in the Data Center Observatory (DCO) serves the dual purposes of providing resources for real users (CMU researchers) and providing us with invaluable Hadoop logs, instrumentation data, and case studies. The logs and data from these systems has been invaluable to our research on problem diagnosis, Big Data tools, cluster resource scheduling, and elastic storage policies. This newsletter and the PDL website offer more details and additional research highlights.

I'm always overwhelmed by the accomplishments of the PDL students and staff, and it's a pleasure to work with them. As always, their accomplishments point at great things to come.



Vivek Seshadri discusses his work on "The Dirty Block Index" with Roger MacNicol of Oracle at the 2014 PDL Spring Industry Visit Day.

Several of Garth's graduate students gather at the 2014 PDL Retreat. From L to R: Swapnil Pimpale, Saurabh Kadekodi, Omkar Gawde, Fiona Britto, and Ravi Chandra Bandlamudi.

YEAR IN REVIEW

May 2015

- 17th annual PDL Spring Visit Day.
- Anuj Kalia presented "Raising the Bar for Using GPUs in Software Packet Processing" at the 12th Usenix Symposium on Networked Systems Design (NSDI'15), in Oakland, CA.

April 2015

- Lavanya Subramanian defended her Ph.D. research on "Providing High and Predictable Performance in Multicore Systems through Shared Resource Management."
- Garth Gibson appointed Associate Dean for CMU's Computer Science masters programs.

March 2015

- Justin Meza received a best presentation award for his talk on "Efficient Data Mapping and Buffering Techniques for Multi-Level Cell Phase-Change Memories" at HiPEAC '15.
- Onur Mutlu received a Google faculty research award.
- Onur Mutlu delivered a keynote talk at the IA3 Workshop on Irregular Applications: Architectures & Algorithms, held during Supercomputing 2014 in New Orleans, LA. His talk was titled, "Rethinking Memory System Design (for Data-Intensive Computing)".
- Lorrie Cranor has been mentioned in several publications for her work in improving individual online security through password alternatives. The San Francisco Gate, InfoWorld and Mercury News each discussed how passwords are no longer sufficient in protecting citizens' online identities, and how biometrics is a possible solution.

February 2015

- Justin Meza received a 2015 Google Ph.D. fellowship for his work on systems reliability.
- "Data Retention in MLC NAND Flash Memory: Characterization, Optimization and Recovery", co-authored by Onur Mutlu and presented by Yu Cai at HPCA-21, was a Best Paper runner-up.

January 2015

- Chad Dougherty was elected CIT's Rookie of the Year!
- Lianghong Xu proposed his Ph.D. thesis research on "Reducing Network Bandwidth for Distributed Document Databases with Similarity-based Deduplication."
- Lorrie Cranor was made an ACM Fellow for her contributions to usable privacy and security.

December 2014

- Peter Klemperer defended his thesis on "Efficient Hypervisor Based Malware Detection."
- Jiaqi Tan presented "STOVE: Strict, Observable, Verifiable Data and Execution Models for Untrusted Applications" at IEEE's 6th International Conference on Cloud Computing Technology and Science (CloudCom) in Singapore.
- Bin Fan presented "Cuckoo Filter: Practically Better Than Bloom" during CoNEXT '14 in Sydney, Australia.

November 2014

- "IndexFS: Scaling File System Metadata Performance with Stateless Caching and Bulk Insertion," presented by Kai Ren at Supercomputing '14 received the Best Paper award.
- Iulian Moraru and his co-authors received the Best Paper award for their paper "Paxos Quorum Leases: Fast Reads Without Sacrificing Writes." Iulian made the presentation at SoCC '14.
- Timothy Zhu completed his SCS speaking skills requirement by presenting "PriorityMeister: Tail Latency QoS for Shared Networked Storage."
- M. Satyanarayanan presented "Cloudlets: at the Leading Edge of Mobile-Cloud Convergence" at MobiCASE 2014: the 6th International Conference on Mobile Computing, Applications and Services, held in Austin, TX.
- Qing Zheng presented "BatchFS: Scaling the File System Control Plane with Client-Funded Metadata Servers" at the 9th International Petascale Data Storage Workshop (PDSW '14), held in conjunction with Supercomputing '14 in New Orleans, LA.
- Timothy Zhu presented "PriorityMeister: Tail Latency QoS for Shared Networked Storage" at the ACM Symposium on Cloud Computing (SoCC'14), in Seattle, WA.
- Henggang Cui presented "Exploiting Iterative-ness for Parallel ML Computations" at the 2014 ACM Symposium on Cloud Computing (SoCC'14), held in Seattle.

October 2014

- Yoongu Kim was the first recipient of Samsung's Ph.D. fellowship.
- Justin Meza proposed his Ph.D. thesis research on "Improving System Reliability with Introspective Hardware/Software Fault Monitoring and Prevention for Memory/Storage Devices."
- Rachata Ausavarungnirun presented "Design and Evaluation of Hierarchical Rings with Deflection Routing" during the 26th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'14) in Paris, France.
- Lavanya Subramanian presented "The Blacklisting Memory Scheduler: Achieving High Performance and Fairness at Low Cost" during the 32nd IEEE International Conference on Computer Design (ICCD) in Seoul, South Korea.
- 22nd annual PDL Retreat.

September 2014

- James Cipar defended his thesis on "Trading Freshness for Performance in Distributed Systems."

continued on page 10

Let's Talk About Storage & Recovery Methods for Non-Volatile Memory Database Systems

Joy Arulraj, Andrew Pavlo & Subramanya R. Dulloor

ACM SIGMOD, Melbourne, Victoria, Australia, May 31-June 4, 2015.

The advent of non-volatile memory (NVM) will fundamentally change the dichotomy between memory and durable storage in database management systems (DBMSs). These new NVM devices are almost as fast as DRAM, but all writes to it are potentially persistent even after power loss. Existing DBMSs are unable to take full advantage of this technology because their internal architectures are predicated on the assumption that memory is volatile. With NVM, many of the components of legacy DBMSs are unnecessary and will degrade the performance of data intensive applications. To better understand these issues, we implemented three engines in a modular DBMS testbed that are based on different storage management architectures: (1) in-place updates, (2) copy-on-write updates, and (3) log-structured updates. We then present NVM-aware variants of these architectures that leverage the persistence and byte-addressability properties of NVM in their storage and recovery methods. Our experimental evaluation on an NVM hardware emulator shows that these engines achieve up to 5.5 higher throughput than their traditional counterparts while reducing the amount of wear due to write operations by up to 2. We also demonstrate that our NVM-aware recovery protocols allow these engines to recover almost instantaneously after the DBMS restarts.

A Cloud Computing Course: From Systems To Services

M. Suhail Rehman, Jason Boles, Mohammad Hammoud & Majd F. Sakr

Proceedings of the 46th ACM Special Interest Group on Computer Science Education Conference (SIGCSE 2015), Kansas City, USA, March 2015.

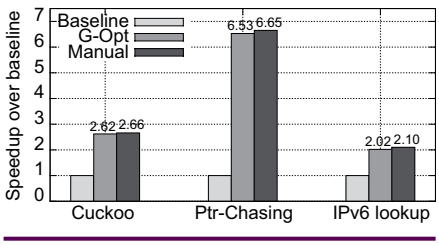
We have designed, developed and administered a course on cloud computing that was taught to over 700 students at our institution over two years. The goal of this project-based course is to provide students with foundational systems concepts as well as experience in developing the required skills to design and deploy viable, robust and elastic web-services within performance and budgetary constraints. We present our objectives, learning outcomes, projects, learning model, outcomes and lessons learned. So far, for this demanding course, our student retention rate is above 80% and enrollment is doubling every year.

Raising the Bar for Using GPUs in Software Packet Processing

Anuj Kalia, Dong Zhou, Michael Kaminsky & David G. Andersen

12th Usenix Symposium on Networked Systems Design (NSDI'15). May 4-6, 2015, Oakland, CA.

Numerous recent research efforts have explored the use of Graphics Processing Units (GPUs) as accelerators for software-based routing and packet handling applications, typically demonstrating throughput several times higher than using legacy code on the CPU alone. In this paper, we explore a new hypothesis about such designs: For many such applications, the benefits arise less from the GPU hardware itself as from the expression of the problem in a language such as CUDA or OpenCL that facilitates memory latency hiding and vectorization through massive concurrency. We demonstrate that in several cases, after applying a similar style of optimization to algorithm implementations, a CPU-only implementation is, in fact, more resource efficient than the version running on the GPU. To "raise the bar" for future uses of GPUs in packet processing applications, we present and evaluate a preliminary language/compiler-based framework called G-Opt that can accelerate CPU-based packet handling programs by automatically hiding memory access latency.



Speedup with G-Opt and manual group prefetching.

Managed Communication and Consistency for Fast Data-Parallel Iterative Analytics

Jinliang Wei, Wei Dai, Aurick Qiao, Qirong Ho, Henggang Cui, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson & Eric P. Xing

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-15-105, April 2015.

At the core of Machine Learning (ML) analytics applied to Big Data is often an expert-suggested model, whose parameters are refined by iteratively processing a training dataset until convergence. The completion time (i.e. convergence time) and quality of the learned model not only depends on the rate at which the refinements are generated but also the quality of each refinement. While data-parallel ML applications often employ a loose consistency model when updating shared model parameters to maximize parallelism, the accumulated error may seriously impact the quality of refinements and thus delay completion time, a problem that usually gets worse with scale. Although more immediate propagation of updates reduces the accumulated error, this strategy is limited by physical network bandwidth. Additionally, the performance of the widely used stochastic gradient

continued on page 6

continued from page 5

descent (SGD) algorithm is sensitive to initial step size, and hand tuning is usually needed to achieve optimal performance.

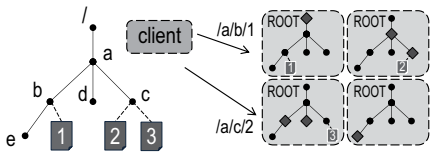
This paper presents Bosen, a system that maximizes the network communication efficiency under a given inter-machine network bandwidth budget to minimize parallel error, while ensuring theoretical convergence guarantees for large-scale data-parallel ML applications. Furthermore, Bosen prioritize messages most significant to algorithm convergence, further enhancing algorithm convergence. Finally, Bosen is the first distributed implementation of the recently presented adaptive revision algorithm, which provides orders of magnitude improvement over a carefully tuned fixed schedule of step size refinements. Experiments on two clusters with up to 1024 cores show that our mechanism significantly improves upon static communication schedules.

ShardFS vs. IndexFS: Replication vs. Caching Strategies for Distributed Metadata Management in Cloud Storage Systems

Lin Xiao, Kai Ren, Qing Zheng & Garth Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-15-104, April 2015.

The rapid growth of cloud storage systems calls for fast and scalable namespace processing. While few commercial file systems offer anything better than federating individually non-scalable namespace servers, a recent academic file system, IndexFS, demonstrates scalable namespace processing based on client caching of directory entries and permissions (directory lookup state) with no per-client state in servers. In this tech report we explore explicit replication of directory lookup state in all servers as an alternative to caching this information in all clients. Both eliminate most repeated RPCs to different servers in order to



ShardFS replicates directory lookup state to all metadata servers so every server can perform path resolution locally. File metadata and non-replicated directory metadata is stored at exactly one server determined by a hash function on the full pathname.

resolve hierarchical permissions tests. Our realization for server replicated directory lookup state, ShardFS, employs a novel file system specific hybrid optimistic and pessimistic concurrency control favoring single object transactions over distributed transactions. Our experimentation suggests that if directory lookup state mutation is a fixed fraction of operations (strong scaling for metadata), server replication does not scale as well as client caching, but if directory lookup state mutation is constant as workload scales (weak scaling for metadata), then server replication can scale more linearly than client caching and provide lower 70 percentile response times as well.

SMPFRAME: A Distributed Framework for Scheduled Model Parallel Machine Learning

Jin Kyu Kim, Qirong Ho, Seunghak Lee, Xun Zheng, Wei Dai, Garth Gibson & Eric Xing

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-15-103, April 2015.

Machine learning (ML) problems commonly applied to big data by existing distributed systems share and update all ML model parameters at each machine using a partition of data — a strategy known as data-parallel. An alternative and complimentary strategy, model-parallel, partitions model parameters for non-shared parallel access and update, periodically repartitioning to facilitate communication. Model-parallelism is motivated by two challenges that data-parallelism does not usually address: (1) parameters may be dependent, thus naive concurrent updates can introduce errors that slow convergence or even cause algorithm failure; (2) model parameters converge at different rates, thus a small subset of parameters can bottleneck ML algorithm completion. We propose scheduled model parallelism (SMP), a programming approach where selection of parameters to be updated (the schedule)is explicitly separated from parameter update logic. The schedule can improve ML algorithm convergence speed by planning for parameter dependencies and uneven convergence. To support SMP at scale, we develop an archetype software framework SMPFRAME which optimizes the throughput of SMP programs, and benchmark four common ML applications written as SMP programs: LDA topic modeling, matrix factorization, sparse least-squares (Lasso) regression and sparse logistic regression. By improving ML progress per iteration through SMP programming whilst improving iteration throughput through SMPFRAME we show that SMP programs running on SMPFRAME outperform non-model-parallel ML implementations: for example, SMP LDA and SMP Lasso respectively achieve 10x and 5x faster convergence than recent, well-established baselines.

Caveat-Scriptor: Write Anywhere Shingled Disks

Saurabh Kadekodi, Swapnil Pimpale & Garth Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-15-101, March, 2015.

Magnetic disks, under pressure from solid state flash storage, are seeking to accelerate the rate that they lower price per stored bit. Magnetic recording technologists have begun to pack tracks

continued on page 7

continued from page 6

so closely that writing one track cannot avoid disturbing the information stored in adjacent tracks. Specifically, the downstream track will be at least partially overwritten, or shingled by each write, and the upstream track will tolerate only a limited number of adjacent writes. Some data that was stored in the downstream track will be lost, forcing firmware or software to ensure that there was no necessary data in parts of that track.

In order to avoid deployment obstacles inherent in asking host software to change before shingled disks are sold, the current generation of shingled disks follow the model established by flash storage: a shingled translation layer of firmware in the disk remaps data writes to empty tracks and cleans (read, move, write) fragmented regions to create empty tracks. Known as Drive-Managed Shingled Disks, host software does not need to change because the disk will do extra work to cope with any write pattern that could lose data. To reduce or eliminate this extra work, changes in the hard disk API have been proposed to enable Host-Managed management of shingled disks.

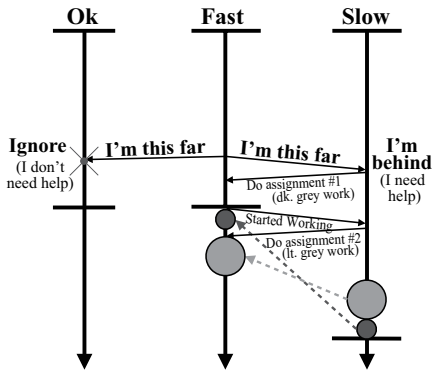
This paper explores two models for Host-Managed Shingled Disk operation. The first, Strict-Append, breaks the disk into fixed sized bands and compels disk writes to occur strictly sequentially in each band, allowing only per-band-truncate-to-empty commands to recover space. This is approximately a physical realization of the classic Log-Structured File System (LFS), and shares the need for the file system to schedule and execute cleaning of bands. The second model, Caveat-Scriptor, exposes a traditional disk address space and a few shingled disks parameters: a distance in the downstream block address space that is guaranteed to never experience shingled overwrite data loss and a distance in the upstream block address space that cannot tolerate multiple adjacent writes. Host-Managed software for Caveat-Scriptor shingled disks is allowed to write anywhere, but if it fails to respect these distance parameters, it may lose data. We show in this paper that Caveat-Scriptor enables reuse of previously written and deleted data with far less cleaning than Strict-Append, enabling the potential for high-density Shingled Disks to perform almost as well as lower-density non-Shingled Disks.

Solving the Straggler Problem for Iterative Convergent Parallel ML

Aaron Harlap, Henggang Cui, Wei Dai, Jinliang Wei, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson & Eric P. Xing

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-15-102. April 2015.

Parallel executions of iterative machine learning (ML) algorithms can suffer significant performance losses to stragglers. The regular (e.g., per iteration) barriers used in the traditional BSP approach cause every transient slowdown of any worker thread to delay all others. This paper describes a scalable, efficient solution to the straggler problem for this important class of parallel ML problems, combining a more flexible synchronization model with dynamic peer-to-peer re-assignment of work among workers. Experiments with both synthetic straggler behaviors and real straggler behavior observed on Amazon EC2 confirm the significance of the problem and the effectiveness of the solution, as implemented in a framework called FlexRR. Using FlexRR, we consistently observe near-ideal runtimes (relative to no performance jitter) across all straggler patterns tested.



Rapid Reassignment example. The middle worker sends progress reports to the other two workers (its helper group). The worker on the left is running at a similar speed, so it ignores the message. The worker on the right is running slower, so it sends a do-this message to re-assign an initial work assignment. Once the faster worker finishes its own work and begins helping, it sends a begun-helping message to the slow worker. Upon receiving this message, the slow worker sends a do-this with a follow-up work assignment to the fast worker.

Data Retention in MLC NAND Flash Memory: Characterization, Optimization and Recovery

Yu Cai, Yixin Luo, Erich F. Haratsch, Ken Mai & Onur Mutlu

HPCA-21, February 7-11, 2015 — Best Paper Runner Up.

Retention errors, caused by charge leakage over time, are the dominant source of flash memory errors. Understanding, characterizing, and reducing retention errors can significantly improve NAND flash memory reliability and endurance. In this paper, we first characterize, with real 2y-nm MLC NAND flash chips, how the threshold voltage distribution of flash memory changes with different retention age – the length of time since a flash cell was programmed. We observe from our characterization results that 1) the optimal read reference voltage of a flash cell, using which the data can be read with the lowest raw bit error rate (RBER), systematically changes with its retention age, and 2) different regions of flash memory can have different retention ages, and hence different optimal read reference voltages. Based on our findings, we propose two new techniques. First, Retention Optimized Reading (ROR) adaptively

continued on page 16

AWARDS & OTHER PDL NEWS

April 2015

Garth Gibson Appointed Associate Dean for Master's Programs

Congratulations to Garth, who has been made Associate Dean for Master's Programs! Dean of Computer Science, Andrew Moore, says "the role of the Associate Dean for Master's Programs is to help coordinate and facilitate the school's mission to produce the very best master's students in the world in our disciplines, and show the ways for other universities to do this. This applies across all types of master's programs: research, academic, and professional. Garth is well-suited to the job: he has a history of making all kinds of organizations successful, and through his co-leadership (along with Eric Nyberg and Majd Sakr) of the Masters in Computational Data Science (founded originally by Anthony Tomasic) has been a very strong contributor within SCS's suite of existing masters programs."

March 2015

PDL Ph.D. Student Receives Best Presentation Award at HiPEAC

Justin Meza received one of the two Best Presentation Awards at the 10th HiPEAC (High Performance and Embedded Architecture and Compilation) conference. The HiPEAC conference is the premier European forum for experts in computer architecture, programming models, compilers and operating systems for embedded and general-purpose systems. The presented paper, titled "Efficient Data Mapping and Buffering Techniques for Multi-Level Cell Phase-Change Memories", is co-authored with ECE's Onur Mutlu, alum HanBin Yoon and researchers from Google.

March 2015

Onur Mutlu Receives Google Faculty Research Award

Congratulations to Onur on receiving a Google Faculty Research Award. Google Research Awards are one-year awards structured as unrestricted gifts to universities to support the work of world-class full-time faculty members at top universities around the world. The intent of the Google Research Awards is to support cutting-edge research in Computer Science, Engineering, and related fields.

This Faculty Award is to support Professor Mutlu's research in the area of novel computer memory systems. Mutlu has been examining new memory architectures and interfaces with the goal of enabling low-cost and energy-efficient computation near data. His related research develops both new hardware substrates and software interfaces to perform computation in or close to memory as well as software techniques that can take better advantage of such new substrates and interfaces. A recent overview of Mutlu's research can be found here.

-- ECE News and google.com

February 2015

Justin Meza Google 2015 PhD Fellowship Recipient

We would like to congratulate Justin Meza, for being selected to receive a Google US/Canada Fellowship for his work in Systems Reliability!

In 2009, Google created the PhD Fellowship program to recognize and support outstanding graduate students doing exceptional work in Computer Science (CS) and related disciplines. In that time we've seen past recipients add depth and breadth to CS by developing new ideas and research directions, from building new intelligence models to changing the way in which we interact with computers to advancing into faculty positions, where they go on to train the next generation of researchers.

-- googleresearch.blogspot.ca, Feb. 18, 2015

January 2015

Chad Dougherty CIT's Rookie of the Year!

The College of Engineering held its annual Staff Awards ceremony last week honoring exceptional staff. Congratulations to Chad Dougherty, a Principle Research Programmer with PDL, on winning CIT's Rookie of the Year Staff Award! Winners of the Rookie award are selected from members who have been a part of CIT for six months to two years as of November 1 in the nomination year, and excellence in the areas of job performance, dedication, positive attitude and contributions as a team player.

January 2015

Lorrie Cranor an ACM Fellow

Congratulations to Lorrie Cranor who has been made an ACM Fellow for contributions to research and education in

continued on page 9

continued from page 8

usable privacy and security.

ACM recognizes its members for contributions to computing that are driving innovations across multiple domains and disciplines. "Our world has been immeasurably improved by the impact of their innovations. We recognize their contributions to the dynamic computing technologies that are making a difference to the study of computer science, the community of computing professionals, and the countless consumers and citizens who are benefiting from their creativity and commitment."

-- ACM Press Room

November 2014

Best Paper at Supercomputing 2014

Congratulations to Kai Ren, Qing Zheng, Swapnil Patil, and Garth Gibson, who received the best paper award at Supercomputing 2014 for their work on "IndexFS: Scaling File System Metadata Performance with Stateless Caching and Bulk Insertion." The paper was chosen from among 84 papers and 394 submissions; the conference hosted 10160 attendees.

The paper, slides and code release are available at http://www.pdl.cmu.edu/indexfs.

November 2014

Best Paper at SoCC!

Congratulations are due to recent PDL Graduate Iulian Moraru and his co-authors David Andersen and Michael Kaminsky for their work on "Paxos Quorum Leases: Fast Reads Without Sacrificing Writes," which received the best paper award at the 2014 ACM Symposium on Cloud Computing (SoCC 2014).

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

Andy 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. 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.

continued on page 10

AWARDS & OTHER PDL NEWS

continued from page 9

"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.

--INI News at www.ini.cmu.edu/news/

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 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.

IN MEMORIAM

Wittatwat Tantisiriroj

We are sad to announce that PDL Alum Wittawat Tantisiriroj tragically lost his life in a car accident on December 17, 2014. He was living in Thailand and working for NECTEC Technologies at the time of his death.

While with the PDL, Wittawat studied storage, databases, and optimization for computing in a massively distributed environment, advised by Garth. He was involved in several research projects, including the Hadoop Distributed Files System (HDFS), Parallel Virtual File Systems (PVFS), DiskReduce, and Scalable Metadata Services.

A friend said, "He had that friendly spark inside him, and shared his optimism and, I think, wonder for the future, with everyone." He is missed.

YEAR IN REVIEW

continued from page 4

August 2014

- Anuj Kalia presented "Using RDMA Efficiently for Key-Value Services" at ACM SIGCOMM 2014, held in Chicago, IL.

July 2014

- Andy Pavlo received the 2014 SIGMOD Jim Gray Doctoral Dissertation Award for his research titled "On Scalable Transaction Execution in Partitioned Main Memory Database Management Systems."
- Iulian Moraru defended his dissertation on "Egalitarian Distributed Consensus."
- Jiaqi Tan presented "CHIPS: Content-based Heuristics for Improving Photo Privacy for Smartphones" at the 7th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec), held in Oxford, UK.

June 2014

- A group of PDL INI students, Aditya Jaltade, Amod Jaltade, Pratik Shah and Mukul Singh, received the Outstanding Student Service Award for a Teaching Assistant for their efforts on the storage systems class taught by Greg and Garth.
- Henggang Cui presented "Exploiting Bounded Staleness to Speed up Big Data Analytics" at the 2014 USENIX Annual Technical Conference (ATC'14), held in Philadelphia, PA.
- Vivek Seshadri presented "The Dirty-Block Index" at the 41st International Symposium on Computer Architecture (ISCA '14) in Minneapolis, MN.
- Samira Khan presented "The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study" at ACM SIGMETRICS 2014 in Austin, TX.

May 2014

- Kevin Kai-Wei Chang, ECE, was awarded the Intel Foundation/SRCEA Graduate Fellowship, in part based on his work on reducing the performance penalty of DRAM refresh, a key limiter of scalability in DRAM memory systems.
- Mukul Singh, INI, submitted his M.S. thesis on "Comparison of Cleaning Performance for SMR Drives."
- Mor Harchol-Balter received two CMU teaching awards.
- 16th annual PDL Spring Visit Day.

continued from page 1

small changes to a database because these devices only support a limited number of writes per address. Shrinking SSDs to smaller sizes also degrades their reliability while increasing interference effects. Stop-gap solutions, such as battery-backed DRAM caches, help mitigate the performance differences but do not resolve these other problems.

Non-volatile memory (NVM) offers an intriguing blend of the two storage mediums. NVM encompasses a broad class of technologies, including phase-change memory, memristors, and STT-MRAM, that can provide low latency reads and writes on the same order of magnitude as DRAM, but with persistent writes and large storage capacity like a SSD.

It is unclear at this point, however, how to best leverage these new technologies in a DBMS. There are several aspects of NVM that make existing DBMS architectures inappropriate for them. For example, disk-oriented DBMSs (e.g., Oracle RDBMS, MySQL) are designed for block-oriented devices that are slow at random access. They maintain an in-memory cache for blocks of tuples and try to maximize the amount of sequential reads and writes to storage. Memory-oriented DBMSs (e.g., VoltDB, MemSQL) contain certain components to overcome the volatility of DRAM, which may be unnecessary in a system with byte-addressable NVM with fast random access.

NVM-Optimized Storage Engines

To better understand these issues, we implemented three storage engines in a modular DBMS test-bed, based on different traditional storage management architectures: (1) in-place updates with logging (InP), (2) copy-on- write updates without logging (CoW), and (3) log-structured updates (Log). We then designed NVM-aware variants of these architectures—NVM-InP, NVM-CoW, and NVM-Log engines—that leverage the persistence and byte-addressability properties of NVM in their storage and recovery methods. For our evaluation, we use a hardware-based emulator where the system only has NVM and volatile CPU-level caches (i.e., no DRAM).

The three traditional engines are designed for a two-tier storage hierarchy comprised of volatile DRAM and a non-volatile HDD/SSD. These storage devices each have distinct hardware constraints and performance properties. First, the read and write latencies of NVM are 2-8 times higher than that of DRAM. The access latencies of NVM are still orders of magnitude

continued on page 12

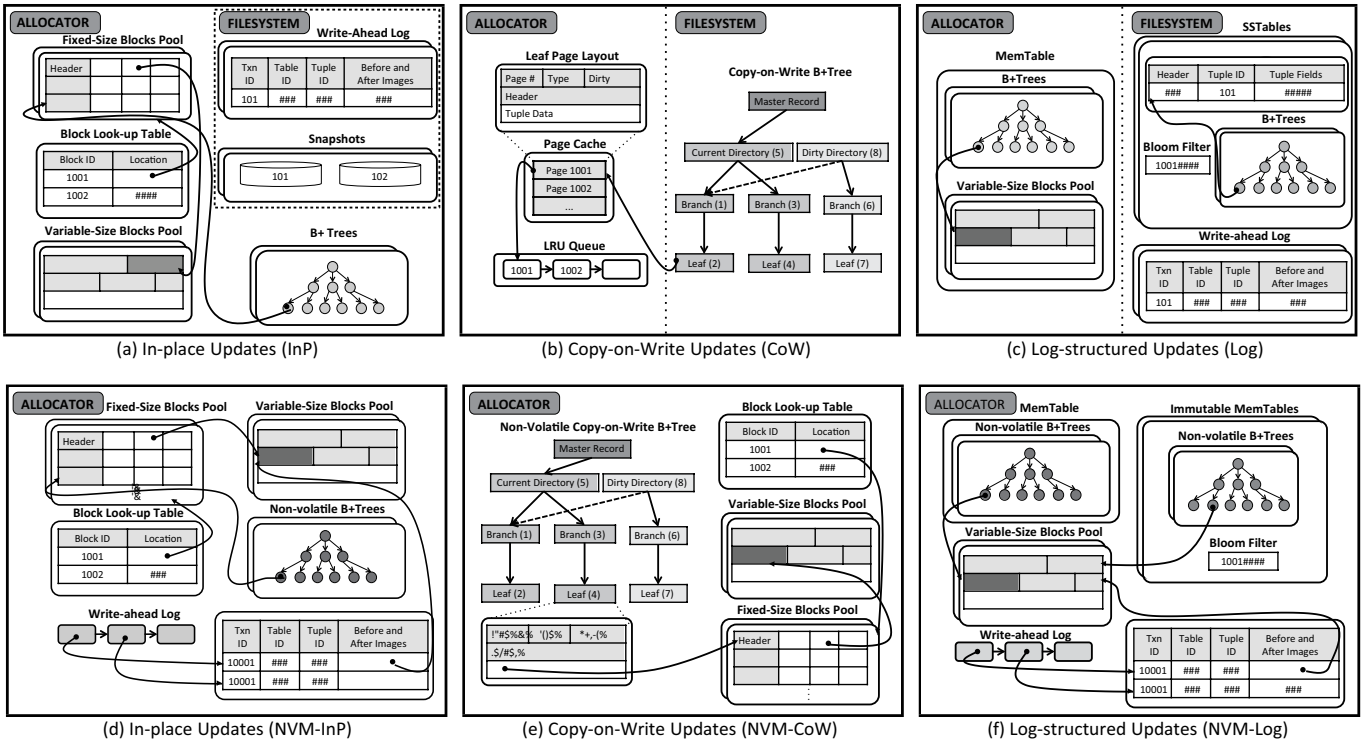


Figure 2: Architectural layout of the three traditional storage engines supported in the DBMS testbed (a, b, and c). The engine components accessed using the allocator interface and those accessed using the filesystem interface are bifurcated by the dashed line. NVM-Aware Engines (d, e, and f)–Architectural layout of the NVM-optimized storage engines.

continued from page 11

shorter than that of HDDs and SSDs. Second, the DBMS can access data on HDD/SSD only at block-granularity, while it can access data stored on NVM at byte-granularity. Further, the performance gap between sequential and random accesses on NVM is comparable to that of DRAM.

The three NVM-optimized storage engines are designed for a single-tier NVM-only storage hierarchy. We designed their storage methods to reduce data duplication. For instance, when a transaction inserts a tuple, rather than copying the tuple to the log, the NVM-InP engine only records a non-volatile tuple pointer in the log. This is sufficient because both the pointer and the tuple referred to by the pointer are stored on NVM. We then developed recovery protocols that allow these engines to recover almost instantaneously after the system restarts by leveraging NVM's persistence property.

Figure 2 compares the architectures of each engine—traditional and modified.

Evaluation

To evaluate the efficacy of our NVM-oriented optimizations, we compared the six different storage engines shown in Figure 2 on the NVM hardware emulator. We based our final evaluation on an analysis of transactional throughput, number of NVM reads/writes, storage footprint, and the time that it takes to recover a database after restarting.

Benchmark: YCSB is a key-value store workload from Yahoo! and is representative of the transactions handled by web-based companies. It contains a single table comprised of tuples with a primary key and 10 columns of random string data, each 100 bytes in size. Each tuple's size is approximately 1 KB. We use a database with 2 million tuples (~2 GB).

Its workload consists of two transaction types: (1) a read transaction that retrieves a single tuple based on its primary key, and (2) an update transaction that modifies a single tuple based on its primary key. Four workload mixtures allowed us to vary the I/O operations that the DBMS executes. For each workload mixture and skew setting pair, we pre-generate a fixed workload of 8 million transactions that is spread evenly across the DBMS's partitions. This allows us to compare the storage footprints and read/write amplification of the engines.

Runtime Performance: First, we analyzed the impact of NVM's latency on the performance of the storage engines. To generalize our analysis to different NVM technologies, we ran the YCSB benchmark under three NVM latency configurations on the hardware emulator: (1) default DRAM latency configuration (160 ns), (2) a low NVM latency configuration that is 2X higher than DRAM latency (320 ns), and (3) a high NVM latency configuration that is 8X higher than DRAM latency (1280 ns).

Reads & Writes: Next, we measured the number of times the storage engines accessed the NVM device while executing the benchmarks using hardware performance counters. These counters track the number of loads from and stores to the NVM device during execution. In each trial, the measurements are collected after loading the initial database.

Recovery: To evaluate the recovery latency of our storage engines we executed a fixed number of transactions and then forced a hard shutdown of the DBMS. The amount of time needed for the system to restore the database to a consistent state was then measured. A consistent state is one where the effects of all committed transactions are durable, and the effects of uncommitted transactions are removed. The number of transactions that need to be recovered by the DBMS depends on the frequency of checkpointing for the InP engine and on the frequency of flushing the MemTable for the Log engine. The CoW and NVM-CoW engines do not perform any recovery mechanism after the OS or DBMS restarts because they never overwrite committed data.

Execution Time Breakdown: In this experiment, we analyzed the time that the engines spent in their internal components during execution using event-based sampling to track the cycles executed within the engine's components. The profiling is started after loading the initial database. The engine's cycles are classified into four categories: (1) storage management operations with the allocator and filesystem interfaces, (2) recovery mechanisms like logging, (3) index accesses and maintenance, and (4) other miscellaneous components. This last category is different for each engine; it includes the time spent in synchronizing the engine's components and performing engine-specific tasks, such as compaction in case of the Log and NVM-Log engines.

Storage Footprint: Finally, we examined the engines' usage of NVM storage at runtime. The storage footprint of an engine is the amount of space that it uses for storing tables, logs, indexes, and other internal data structures. This metric is important because we expect that the first NVM products will initially have a higher cost than current storage technologies. For this experiment, statistics maintained by our allocator and the filesystem meta-data are collected during workload execution, after loading the initial database. We then report the peak storage footprint of each engine.

continued on page 13

continued from page 13

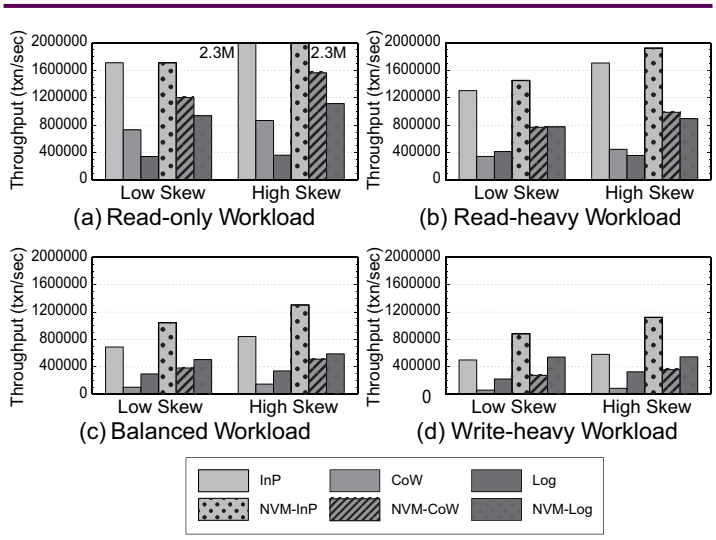


Figure 3: YCSB Performance (Low Latency)—The throughput of the engines for the YCSB benchmark under the low NVM latency configuration (2×).

Discussion

Our analysis showed that the NVM access latency had the most impact on the runtime performance of the engines, more so than the amount of skew or the number of modifications to the database in the workload. The difference due to latency was more pronounced with the NVM-aware variants; their absolute throughput was better than the traditional engines, but longer latencies caused performance to drop more significantly. This is because they are no longer bottlenecked by heavyweight durability mechanisms.

The NVM-aware engines also perform fewer store operations, which will help extend NVM device lifetime. We attribute this to the reduction in redundant data that the engines store when a transaction modifies the database. Using the allocator interface with non-volatile pointers for internal data structures also allows them to have a smaller storage footprint. This in turn avoids polluting the CPU's caches with unnecessary copy and transform operations. It also improves recovery times of engines that use a WAL since they no longer record redo information.

Overall, we found that the NVM-InP engine performed the best across a wide set of workload mixtures and skew settings for all NVM latency configurations. The NVM-CoW engine did not perform as well for write-intensive workloads, but may be a better fit for DBMSs that support non-blocking read-only transactions. Many of the design assumptions made in the NVM-Log engine are not suitable for a single-tier storage hierarchy. This engine is essentially performing in-place updates like the NVM-InP engine but with additional overhead of maintaining its legacy components.

For more details on all of our experiments, please see [1].

Figure 3 provides an example of the data collected in our experiments: YCSB Performance (Low Latency) – The throughput of the engines for the YCSB benchmark under the low NVM latency configuration (2X).

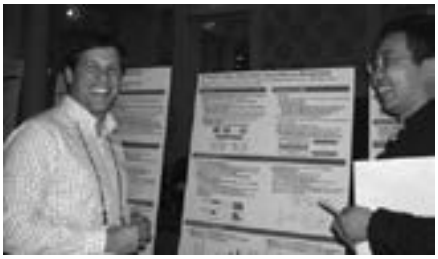
Conclusions & Future Work

Our experimental evaluation on an NVM hardware emulator showed that the NVM-optimized engines deliver up to 5.5X higher throughput than their traditional counterparts, while reducing the amount of wear due to write operations by up to 2X. NVM-aware recovery protocols allow these engines to recover almost instantaneously after the DBMS restarts. We found that the NVM access latency has the most impact on the runtime performance of the engines, more so than the workload skew or the number of modifications to the database in the workload. Our evaluation showed that the NVM-aware in-place updates engine achieved the best throughput among all the engines with the least amount of wear on the NVM device.

We found that a hybrid DRAM and NVM storage hierarchy is a viable alternative to traditionally managed DBMS systems, particularly in case of high NVM latency technologies and analytical workloads. In future work, we plan to optimize our NVM-aware engines for such a hybrid storage hierarchy. We are also interested in exploring methods for supporting hybrid workloads (i.e., OLTP + OLAP) on NVM. We anticipate the need for techniques to protect the contents of the database stored on NVM from errant code running.

References

[1] Let's Talk About Storage & Recovery Methods for Non-Volatile Memory Database Systems. Joy Arulraj, Andrew Pavlo, Subramanya R. Dulloor. Proceedings ACM SIGMOD, Melbourne, Victoria, Australia, May 31-June 4, 2015.



Justin Levandoski (Microsoft Research) and Yang Li (CMU) discuss Yang's research on "A Case For Page Utility Based Hybrid Memory Management" at one of the PDL Retreat poster sessions.

DISSERTATION ABSTRACT:
Efficient Hypervisor Based Malware Detection

Peter Friedrich Klemperer

Carnegie Mellon University, ECE

Ph.D. Dissertation, May 2015

Recent years have seen an uptick in master boot record (MBR) based rootkits that load before the Windows operating system and subvert the operating system's own procedures. As such, MBR rootkits are difficult to counter with operating system-based antivirus software that runs at the same privilege-level as the rookits. Hypervisors operate at a higher privilege level than the guests they manage, creating a high-ground position in the host. This high-ground position can be exploited to perform security checks on the virtual machine guests where the checking software is isolated from guest-based viruses. The efficient introspection system described in this thesis targets existing virtualized systems to improve security with real-time, concurrent memory introspection capabilities. Efficient introspection decouples memory introspection from virtual machine guest execution, establishes coherent and consistent memory views between the host and running guest, while maintaining normal guest operation. Existing introspection systems have provided one or two of these properties but not all three at once.

Jin Kyu Kim describes his research on "STRADS: Scheduling for Parallel Machine Learning" at the 2014 Parallel Data Lab Retreat.

This thesis presents a new concurrent-computing approach – high-performance memory snapshotting – to accelerating hypervisor based introspection of virtual machine guest memory that combines all three elements to improve performance and security. Memory snapshots create a coherent and consistent memory view of the guest that can be shared with the independently running introspection application. Three memory snapshotting mechanisms are presented and evaluated for their impact on normal guest operation.

Existing introspection systems and security protection techniques that were previously dismissed as too slow are now be enabled by efficient introspection. This thesis explains why existing introspection systems are inadequate, describes how existing system performance can be improved, evaluates an efficient introspection prototype on both applications and microbenchmarks, and discusses two potential security applications that are enabled by efficient introspection. These applications point to efficient introspection's utility for supporting useful security applications.

DISSERTATION ABSTRACT:
Trading Freshness for Performance in Distributed Systems

James Cipar

Carnegie Mellon University, SCS

Ph.D. Dissertation, December 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 a 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 originally designed for to store meta-data extracted from enterprise computer files, for the purposes of enterprise information management. 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.

continued on page 15

continued from page 14

DISSERTATION ABSTRACT:
Egalitarian Distributed Consensus

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 theoretically 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.

Wolf Richter speaks about his research on "Agentless Cloud-wide Monitoring of Virtual Disk State" at the 2014 PDL Retreat.

THESIS PROPOSAL:
Reducing Network Bandwidth for Distributed Document Databases with Similarity-based Deduplication

Lianghong Xu, ECE

January 21, 2015

With the rise of large-scale, Web-based applications, users are increasingly adopting a new class of document-oriented database management systems (DBMSs) that allow for rapid prototyping while also achieving scalable performance. Like for other distributed storage systems, replication is an important consideration for document DBMSs in order to guarantee availability. Replication can be between failure-independent nodes in the same data center and/or in geographically diverse data centers. A replicated DBMS maintains synchronization across multiple nodes by sending operation logs (oplogs) across the network, and the network bandwidth required can become a bottleneck. As such, there is a strong need to reduce the bandwidth required to maintain secondary database replicas, especially for geo-replication scenarios where wide-area network (WAN) bandwidth is expensive and capacities grow slowly across infrastructure upgrades over time.

This work presents a deduplication system called sDedup that reduces the amount of data transferred over the network for replicated document DBMSs. sDedup uses similarity-based deduplication to remove redundancy in oplog entries by delta encoding against similar documents selected from the entire database. It exploits key workload characteristics of document-oriented workloads, including small item sizes, temporal locality, and incremental nature of document edits. Our experimental evaluation of sDedup using MongoDB with three real-world datasets shows that it is able to achieve up to 38X reduction in oplog bytes sent over the network, in addition to the standard 3X reduction from compression, significantly outperforming traditional chunk-based deduplication techniques while incurring negligible performance overhead.

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

continued on page 16

DISSERTATIONS & PROPOSALS

continued from page 15

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.

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 xed 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.

RECENT PUBLICATIONS

continued from page 7

learns and applies the optimal read reference voltage for each flash memory block online. The key idea of ROR is to periodically learn a tight upper bound, and from there approach the optimal read reference voltage. Our evaluations show that ROR can extend flash memory lifetime by 64% and reduce average error correction latency by 10.1%, with only 768 KB storage overhead in flash memory for a 512 GB flash-based SSD. Second, Retention Failure Recovery (RFR) recovers data with uncorrectable errors offline by identifying and probabilistically correcting flash cells with retention errors. Our evaluation shows that RFR reduces RBER by 50%, which essentially doubles the error correction capability, and thus can effectively recover data from otherwise uncorrectable flash errors.

Mitigating Prefetcher-Caused Pollution Using Informed Caching Policies for Prefetched Blocks

Vivek Seshadri, Samihan Yedkar, Hongyi Xin, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch & Todd C. Mowry

ACM Transactions on Architecture and Code Optimization (TACO), Volume 11 Issue 4, January 2015, Article No. 51.

Many modern high-performance processors prefetch blocks into the on-chip cache. Prefetched blocks can potentially pollute the cache by evicting more useful blocks. In this work, we observe that both accurate and inaccurate prefetches lead to cache pollution, and propose a comprehensive mechanism to mitigate prefetcher-caused cache pollution.

First, we observe that over 95% of useful prefetches in a wide variety of applications are not reused after the first demand hit (in secondary caches). Based on this observation, our first mechanism simply demotes a prefetched block to the lowest priority on a demand hit. Second, to address pollution caused by inaccurate prefetches, we propose a self-tuning prefetch accuracy predictor to predict if a prefetch is accurate or inaccurate. Only predicted-accurate

continued on page 17

continued from page 16

prefetches are inserted into the cache with a high priority.

Evaluations show that our final mechanism, which combines these two ideas, significantly improves performance compared to both the baseline LRU policy and two state-of-the-art approaches to mitigating prefetcher-caused cache pollution (up to 49%, and 6% on average for 157 two-core multiprogrammed workloads). The performance improvement is consistent across a wide variety of system configurations.

The Main Memory System: Challenges and Opportunities

Onur Mutlu, Justin Meza & Lavanya Subramanian

Invited Article in Communications of the Korean Institute of Information Scientists and Engineers (KIISE), 2015.

The memory system is a fundamental performance and energy bottleneck in almost all computing systems. Recent system design, application, and technology trends that require more capacity, bandwidth, efficiency, and predictability out of the memory system make it an even more important system bottleneck. At the same time, DRAM technology is experiencing difficult technology scaling challenges that make the maintenance and enhancement of its capacity, energy-efficiency, and reliability significantly more costly with conventional techniques.

In this article, after describing the demands and challenges faced by the memory system, we examine some promising research and design directions to overcome challenges posed by memory scaling. Specifically, we describe three major new research challenges and solution directions: 1) enabling new DRAM architectures, functions, interfaces, and better integration of the DRAM and the rest of the system (an approach we call system-DRAM co-design), 2) designing a memory system that employs emerging non-volatile memory technologies and takes advantage of multiple different technologies (i.e., hybrid memory systems), 3) providing predictable performance and QoS to applications sharing the memory system (i.e., QoS-aware memory systems). We also briefly describe our ongoing related work in combating scaling challenges of NAND flash memory.

STOVE: Strict, Observable, Verifiable Data and Execution Models for Untrusted Applications

Jiaqi Tan, Rajeev Gandhi & Priya Narasimhan

IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom), 2014 (Doctoral Symposium), pp.644,649, 15-18 Dec. 2014.

The massive growth in mobile devices is likely to give rise to the leasing out of compute and data resources on mobile devices to third-parties to enable applications to be run across multiple mobile devices. However, users who lease their mobile devices out need to run applications from unknown third parties, and these untrusted applications may harm their devices or access unauthorized personal data.

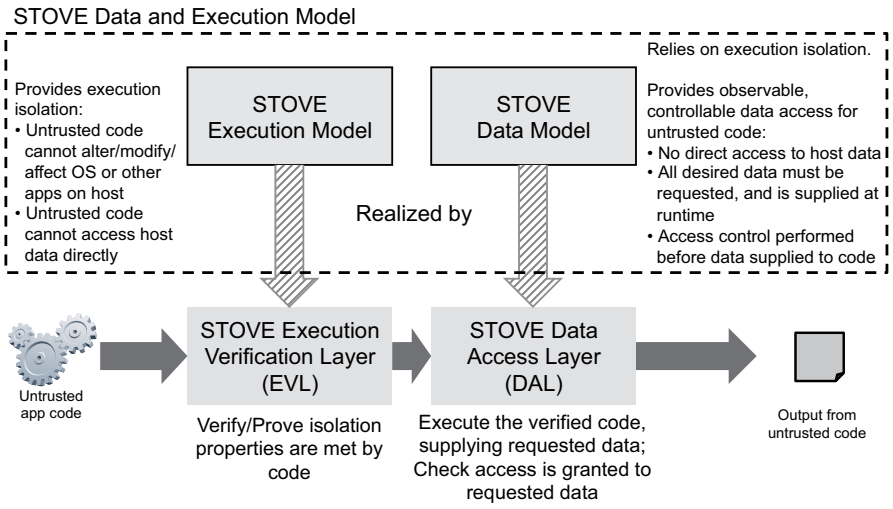
We propose STOVE, a data and execution model for structuring untrusted applications to be secure by construction, to achieve strict and verifiable execution isolation, and observable access control for data. STOVE uses formal logic to verify that untrusted code meets isolation properties which imply that hosts running the code cannot be harmed, and that untrusted code cannot directly access host data. STOVE performs all data accesses on behalf of untrusted code, allowing all access control decisions to be reliably performed in one place. Thus, users can run untrusted applications structured using the STOVE model on their systems, with strong guarantees, based on formal proofs, that these applications will not harm their system nor access unauthorized data.

A Comparative Study of Baremetal Provisioning Frameworks

Ashok Chandrasekar & Garth Gibson

Carnegie Mellon University Parallel Data Laboratory Technical Report CMU-PDL-14-109, December 2014.

There are many baremetal frameworks available in the market today. These



Overview of the STOVE Data and Execution Model, and our approach.

continued on page 18

continued from page 16

baremetal frameworks help ease the work of a cluster/datacenter administrator by automating the deployment of operating systems and other software onto the individual machines in a datacenter. These frameworks differ from each other on several aspects like price, stability, maintainability, feature support and performance. As a result, it becomes difficult to choose the best framework to suit one's needs. This study aims at helping the administrator in choosing the appropriate framework based on their needs by providing a comparison of these frameworks with a main emphasis on Emulab and OpenStack Ironic.

Efficient Data Mapping and Buffering Techniques for Multilevel Cell Phase-Change Memories

Hanbin Yoon, Justin Meza, Naveen Mural Imanohar, Norman P. Jouppi & Onur Mutlu

ACM Transactions on Architecture and Code Optimization, Vol. 11, No. 4, Article 40, December 2014.

New phase-change memory (PCM) devices have low-access latencies (like DRAM) and high capacities (i.e., low cost per bit, like Flash). In addition to being able to scale to smaller cell sizes than DRAM, a PCM cell can also store multiple bits per cell (referred to as multilevel cell, or MLC), enabling even greater capacity per bit. However, reading and writing the different bits of data from and to an MLC PCM cell requires different amounts of time: one bit is read or written first, followed by another. Due to this asymmetric access process, the bits in an MLC PCM cell have different access latency and energy depending on which bit in the cell is being read or written.

We leverage this observation to design a new way to store and buffer data in MLC PCM devices. While traditional devices couple the bits in each cell next to one another in the address space, our key idea is to logically decouple the bits in each cell into two separate regions depending on their read/write characteristics: fast-read/slow-write bits and slow-read/fast-write bits. We propose a low-overhead hardware/software technique to predict and map data that would benefit from being in each region at runtime. In addition, we show how MLC bit decoupling provides more flexibility in the way data is buffered in the device, enabling more efficient use of existing device buffer space.

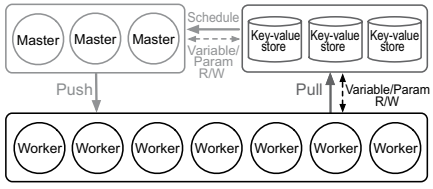
Our evaluations for a multicore system show that MLC bit decoupling improves system performance by 19.2%, memory energy efficiency by 14.4%, and thread fairness by 19.3% over a state-of-the-art MLC PCM system that couples the bits in its cells. We show that our results are consistent across a variety of workloads and system configurations.

On Model Parallelization and Scheduling Strategies for Distributed Machine Learning

S. Lee, J. K. Kim, X. Zheng, Q. Ho, G. A. Gibson & E. P. Xing

Proceedings of 2014 Neural Information Processing Systems (NIPS'14), December 2014.

Distributed machine learning has typically been approached from a data parallel perspective, where big data are partitioned to multiple workers and an algorithm is executed concurrently over different data subsets under various synchronization schemes to ensure speed-up and/or correctness. A sibling problem that has received relatively less attention is how to ensure efficient and correct model parallel execution of ML algorithms, where parameters of an ML program are partitioned to different workers and undergone concurrent iterative updates. We argue that model and data parallelisms impose rather different challenges for system design, algorithmic adjustment, and theoretical analysis. In this paper, we develop a system for model-parallelism, STRADS, that provides a programming abstraction for scheduling parameter updates by discovering and leveraging changing structural properties of ML programs. STRADS enables a flexible tradeoff between scheduling efficiency and fidelity to intrinsic dependencies within the models, and improves memory efficiency of distributed ML. We demonstrate the efficacy of model-parallel algorithms implemented on STRADS versus popular implementations for topic modeling, matrix factorization, and Lasso.



High-level architecture of our STRADS system interface for dynamic model parallelism.

Cuckoo Filter: Practically Better Than Bloom

Bin Fan, David G. Andersen, Michael Kaminsky & Michael D. Mitzenmacher

Proceedings of CoNEXT (CoNEXT'14), December 2014.

In many networking systems, Bloom filters are used for highspeed set membership tests. They permit a small fraction of false positive answers with very good space efficiency. However, they do not permit deletion of items from the set, and previous attempts to extend "standard" Bloom filters to support deletion all degrade either space or performance.

We propose a new data structure called the cuckoo filter that can replace Bloom filters for approximate set membership tests. Cuckoo filters support adding and removing items dynamically while achieving even higher performance than Bloom filters. For applications that store many items and

continued on page 19

continued from page 18

target moderately low false positive rates, cuckoo filters have lower space overhead than space-optimized Bloom filters. Our experimental results also show that cuckoo filters outperform previous data structures that extend Bloom filters to support deletions substantially in both time and space.

Reducing Replication Bandwidth for Distributed Document Databases

Lianghong Xu, Andrew Pavlo, Sudipta Sengupta Jin Li & Gregory R. Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-14-108. December 2014.

With the rise of large-scale, Web-based applications, users are increasingly adopting a new class of document-oriented database management systems (DBMSs) that allow for rapid prototyping while also achieving scalable performance. Like for other distributed storage systems, replication is an important consideration for document DBMSs in order to guarantee availability. Replication can be between failure-independent nodes in the same data center and/or in geographically diverse data centers. A replicated DBMS maintains synchronization across multiple nodes by sending operation logs (oplogs) across the network, and the network bandwidth required can become a bottleneck. As such, there is a strong need to reduce the bandwidth required to maintain secondary database replicas, especially for geo-replication scenarios where wide-area network (WAN) bandwidth is expensive and capacities grow slowly across infrastructure upgrades over time.

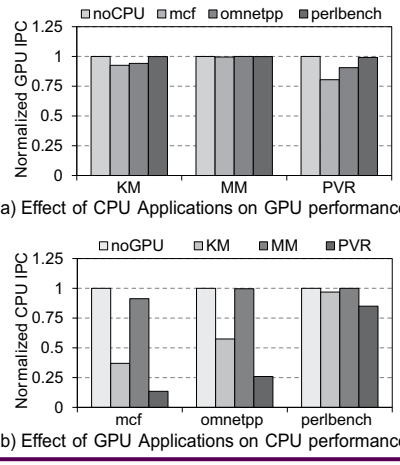
This paper presents a deduplication system called sDedup that reduces the amount of data transferred over the network for replicated document DBMSs. sDedup uses similarity-based deduplication to remove redundancy of documents in oplog entries by delta encoding against similar documents selected from the entire database. It exploits key workload characteristics of document-oriented workloads, including small document sizes, temporal locality, and incremental nature of document edits. Our experimental evaluation of sDedup using MongoDB with three real-world datasets shows that it is able to achieve up to 38 reduction in oplog bytes sent over the network, in addition to the standard 3X reduction from compression, significantly outperforming traditional chunk-based deduplication techniques while incurring negligible performance overhead.

Managing GPU Concurrency in Heterogeneous Architectures

Onur Kayiran, Nachiappan Chidambaram, Adwait Jog, Rachata Ausavarungnirun, Mahmut T. Kandemir, Gabriel H. Loh, Onur Mutlu & Chita R. Das

Proceedings of 47th International Symposium on Microarchitecture (MICRO'14), December 2014.

Heterogeneous architectures consisting of general purpose CPUs and throughput-optimized GPUs are projected to be the dominant computing platforms for many classes of applications. The design of such systems is more complex than that of homogeneous architectures because maximizing resource utilization while minimizing shared resource interference between CPU and GPU applications is difficult. We show that GPU applications tend to monopolize the shared hardware resources, such as memory and network, because of their high thread-level parallelism (TLP), and discuss the limitations of existing GPU-based concurrency management techniques when employed in heterogeneous systems. To solve this problem, we propose an integrated concurrency management strategy that modulates the TLP in GPUs to control the performance of both CPU and GPU applications.



Effects of heterogeneous execution on performance.

This mechanism considers both GPU core state and system-wide memory and network congestion information to dynamically decide on the level of GPU concurrency to maximize system performance. We propose and evaluate two schemes: one (CM-CPU) for boosting CPU performance in the presence of GPU interference, the other (CM-BAL) for improving both CPU and GPU performance in a balanced manner and thus overall system performance. Our evaluations show that the first scheme improves average CPU performance by 24%, while reducing average GPU performance by 11%. The second scheme provides 7% average performance improvement for both CPU and GPU applications. We also show that our solution allows the user to control performance trade-offs between CPUs and GPUs.

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, persis-

continued on page 20

continued from page 19

tent 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 under-utilization 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.

Cloudlets: at the Leading Edge of Mobile-Cloud Convergence

M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter & P. Pillai

Proceedings of MobiCASE 2014: Sixth International Conference on Mobile Computing, Applications and Services, Austin, TX, November 2014.

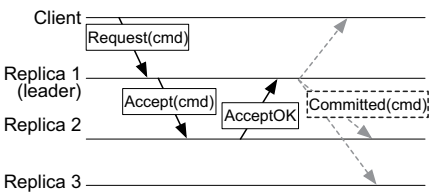
As mobile computing and cloud computing converge, the sensing and interaction capabilities of mobile devices can be seamlessly fused with compute-intensive and data-intensive processing in the cloud. Cloudlets are important architectural components in this convergence, representing the middle tier of a mobile device — cloudlet — cloud hierarchy. We show how cloudlets enable a new genre of applications called cognitive assistance applications that augment human perception and cognition. We describe a plug-and-play architecture for cognitive assistance, and a proof of concept using Google Glass.

Paxos Quorum Leases: Fast Reads Without Sacrificing Writes

Iulian Moraru, David G. Andersen & Michael Kaminsky

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

This paper describes quorum leases, a new technique that allows Paxos-based systems to perform reads with high throughput and low latency. Quorum leases do not sacrifice consistency and have only a small impact on system availability and write latency. Quorum leases allow a majority of replicas to perform strongly consistent local reads, which substantially reduces read latency at those replicas (e.g., by two orders of magnitude in wide-area scenarios). Previous techniques for performing local reads in Paxos systems either (a) sacrifice consistency; (b) allow only one replica to read locally; or (c) decrease the availability of



Steady state interaction in Multi-Paxos. The asynchronous messages are represented as dashed arrows.

the system and increase the latency of all updates by requiring all replicas to be notified synchronously. We describe the design of quorum leases and evaluate their benefits compared to previous approaches through an implementation running in five geo-distributed Amazon EC2 datacenters.

BatchFS: Scaling the File System Control Plane with Client-Funded Metadata Servers

Qing Zheng, Kai Ren & Garth Gibson

Proceedings of the 9th international Petascale Data Storage Workshop (PDSW '14) held in conjunction with Supercomputing '14. November 16, 2014, New Orleans, LA.

Parallel file systems are often characterized by a layered architecture that decouples metadata management from I/O operations, allowing file systems to facilitate fast concurrent access to file contents. However, metadata intensive workloads are still likely to bottleneck at the file system control plane due to namespace synchronization, which taxes application performance through lock contention on directories, transaction serialization, and RPC overheads. In this paper, we propose a client-driven file system metadata architecture, BatchFS, that is optimized for non-interactive, or batch, workloads. To avoid metadata bottlenecks, BatchFS features a relaxed consistency model marked by lazy namespace synchronization and

continued on page 21

continued from page 20

optimistic metadata verification. Capable of executing namespace operations on client-provisioned resources without contacting any metadata server, BatchFS clients are able to delay namespace synchronization until synchronization is really needed. Our goal in this vision paper is to handle these delayed operations securely and efficiently with metadata verification and bulk insertion. Preliminary experiments demonstrate that our client-funded metadata architecture outperforms a traditional synchronous file system by orders of magnitude.

A Brief History of Cloud Offload

M. Satyanarayanan

GetMobile, Volume 18, Issue 4, October 2014.

Every time you use a voice command on your smartphone, you are benefitting from a technique called cloud offload. Your speech is captured by a microphone, pre-processed, then sent over a wireless network to a cloud service that converts speech to text. The result is then forwarded to another cloud service or sent back to your mobile device, depending on the application. Speech recognition and many other resource-intensive mobile services require cloud offload. Otherwise, the service would be too slow and drain too much of your battery.

The Heterogeneous Block Architecture

Chris Fallin, Chris Wilkerson & Onur Mutlu

Proceedings of 32nd IEEE International Conference on Computer Design (ICCD'14), October 2014.

This paper makes two observations that lead to a new heterogeneous core design. First, we observe that most serial code exhibits fine-grained heterogeneity: at the scale of tens or hundreds of instructions, regions of code fit different microarchitectures better (at the same point or at different points in time). Second, we observe that by grouping contiguous regions of instructions into blocks that are executed atomically, a core can exploit this fine-grained heterogeneity: atomicity allows each block to be executed independently on its own execution backend that fits its characteristics best.

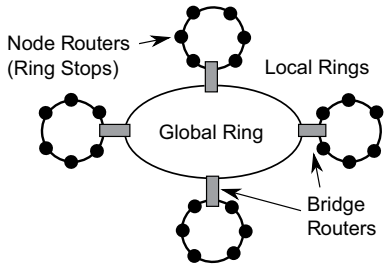
Based on these observations, we propose a fine-grained heterogeneous core design, called the heterogeneous block architecture (HBA), that combines heterogeneous execution backends into one core. HBA breaks the program into blocks of code, determines the best backend for each block, and specializes the block for that backend. As an example HBA design, we combine out-of-order, VLIW, and in-order backends, using simple heuristics to choose backends for different dynamic instruction blocks. Our extensive evaluations compare this example HBA design to multiple baseline core designs (including monolithic out-of-order, clustered out-of-order, in-order and a state-of-the-art heterogeneous core design) and show that it provides significantly better energy efficiency than all designs at similar performance.

Design and Evaluation of Hierarchical Rings with Deflection Routing

Rachata Ausavarungnirun, Chris Fallin, Xiangyao Yu, Kevin Chang, Greg Nazario, Reetuparna Das, Gabriel Loh & Onur Mutlu

Proceedings of the 26th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'14), October 2014.

Hierarchical ring networks, which hierarchically connect multiple levels of rings, have been proposed in the past to improve the scalability of ring interconnects, but past hierarchical ring designs sacrifice some of the key benefits of rings by reintroducing more complex in-ring buffering and buffered flow control. Our goal in this paper is to design a new hierarchical ring interconnect that can maintain most of the simplicity of traditional ring designs (i.e., no in-ring buffering or buffered flow control) while achieving high scalability as more complex buffered hierarchical ring designs.



A traditional hierarchical ring design allows "local rings" with simple node routers to scale by connecting to a "global ring" via bridge routers.

To this end, we revisit the concept of a hierarchical-ring network--on-chip. Our design, called HiRD (Hierarchical Rings with Deflection), includes critical features that enable us to mostly maintain the simplicity of traditional simple ring topologies while providing higher energy efficiency and scalability. First, HiRD does not have any buffering or buffered flow control within individual rings, and requires only a small amount of buffering between the ring hierarchy levels. When inter-ring buffers are full, our design simply deflects flits so that they circle the ring and try again, which eliminates the need for in-ring buffering. Second, we introduce two simple mechanisms that together provide an end-to-end delivery guarantee within the entire network (despite any deflections that occur) without impacting the critical path or latency of the vast majority of network traffic.

Our experimental evaluations on a wide variety of multiprogrammed and multithreaded workloads and synthetic traffic patterns show that HiRD at-

continued on page 22

continued from page 21

tains equal or better performance at better energy efficiency than multiple versions of both a previous hierarchical ring design and a traditional single ring design. We also extensively analyze our design's characteristics and injection and delivery guarantees. We conclude that HiRD can be a compelling design point that allows higher energy efficiency and scalability while retaining the simplicity and appeal of conventional ring-based designs.

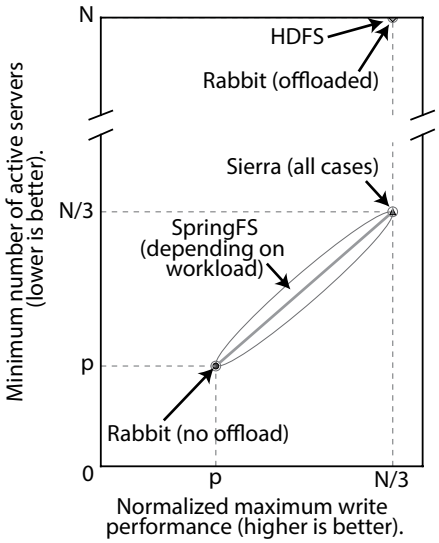
Agility and Performance in Elastic Distributed Storage

Lianghong Xu, James Cipar, Elie Krevat, Alexey Tumanov, And Nitin Gupta, Michael A. Kozuch, Gregory R. Ganger

ACM Transactions on Storage, Vol. 10, No. 4, Article 16, October 2014.

Elastic storage systems can be expanded or contracted to meet current demand, allowing servers to be turned off or used for other tasks. However, the usefulness of an elastic distributed storage system is limited by its agility: how quickly it can increase or decrease its number of servers. Due to the large amount of data they must migrate during elastic resizing, state of the art designs usually have to make painful trade-offs among performance, elasticity, and agility. This article describes the state of the art in elastic storage and a new system, called SpringFS, that can quickly change its number of active servers, while retaining elasticity and performance goals. SpringFS uses a novel technique, termed bounded write offloading, that restricts the set of servers where writes to overloaded servers are redirected.

This technique, combined with the read offloading and passive migration policies used in SpringFS, minimizes the work needed before deactivation or activation of servers. Analysis of real-world traces from Hadoop deployments at Facebook and various Cloudera customers and experiments with the SpringFS prototype confirm



Elastic storage system comparison in terms of agility and performance. *N* is the total size of the cluster. *p* is the number of primary servers in the equal-work data layout. Servers with at least some primary replicas cannot be deactivated without first moving those primary replicas. SpringFS provides a continuum between Sierra's and Rabbit's (when no offload) single points in this trade-off space. When Rabbit requires offload, SpringFS is superior at all points. Note that the y-axis is discontinuous.

SpringFS's agility, show that it reduces the amount of data migrated for elastic resizing by up to two orders of magnitude, and show that it cuts the percentage of active servers required by 67–82%, outdoing state-of-the-art designs by 6–120%.

Fast and Accurate Mapping of Complete Genomics Reads

Donghyuk Lee, Farhad Hormozdiari, Hongyi Xin, Faraz Hach & Onur Mutlu, Can Alkan

Methods, Elsevier, October 2014.

Many recent advances in genomics and the expectations of personalized medicine are made possible thanks to power of high throughput sequencing (HTS) in sequencing large collections of human genomes. There are tens of different sequencing technologies currently available, and each HTS platform have different strengths and biases. This diversity both makes it possible to use different technologies to correct for shortcomings; but also requires to develop different algorithms for each platform due to the differences in data types and error models. The first problem to tackle in analyzing HTS data for resequencing applications is the read mapping stage, where many tools have been developed for the most popular HTS methods, but publicly available and open source aligners are still lacking for the Complete Genomics (CG) platform. Unfortunately, Burrows-Wheeler based methods are not practical for CG data due to the gapped nature of the reads generated by this method. Here we provide a sensitive read mapper (sirFAST) for the CG technology based on the seed-and-extend paradigm that can quickly map CG reads to a reference genome. We evaluate the performance and accuracy of sirFAST using both simulated and publicly available real data sets, showing high precision and recall rates.

Value Driven Load Balancing

Sherwin Doroudi, Esa Hyytia & Mor Harchol-Balter

Performance Evaluation, vol. 79, September 2014.

To date, the study of dispatching or load balancing in server farms has primarily focused on the minimization of response time. Server farms are typically modeled by a front-end router that employs a dispatching policy to route jobs to one of several servers, with each server scheduling all the jobs in its queue via Processor-Sharing. However, the common assumption has been that all jobs are equally important or valuable, in that they are equally sensitive to delay. Our work departs from this assumption: we model each arrival as having a randomly distributed value parameter,

continued on page 23

continued from page 22

independent of the arrival's service requirement (job size). Given such value heterogeneity, the correct metric is no longer the minimization or response time, but rather, the minimization of value-weighted response time. In this context, we ask "what is a good dispatching policy to minimize the value-weighted response time metric?" We propose a number of new dispatching policies that are motivated by the goal of minimizing the value-weighted response time. Via a combination of exact analysis, asymptotic analysis, and simulation, we are able to deduce many unexpected results regarding dispatching.

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.

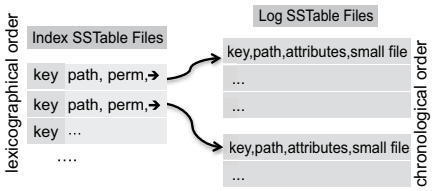
Tail latency service level objectives (SLOs) are an important, but very challenging, problem for cloud computing infrastructures. Existing approaches are effective for average-case performance and low-burstiness workloads, but not for tail latency SLOs under bursty workloads. This paper describes PriorityMeister, 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 misestimation of underlying storage device performance and containing the effect of misbehaving workloads.

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 storm-free 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.



Column-style stores index and log tables separately. Index tables contain frequently accessed attributes for file lookups and a pointer to the location of full file metadata in the most recent log file. Index tables are compacted while log tables are not, reducing the total work for IndexFS.

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 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

continued on page 24

continued from page 23

improvements are robust to variation in the patterns over time.

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.

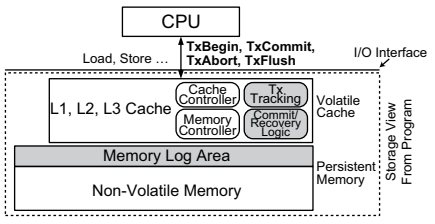
Loose-Ordering Consistency for Persistent Memory

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.

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 overall system performance and unfairly slows down applications. Prior works have developed application-aware memory request schedulers to

continued on page 25

continued from page 24

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).

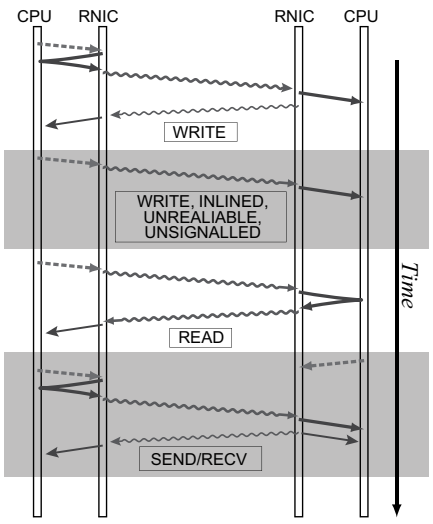
Using RDMA Efficiently for Key-Value Services

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 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.



Steps involved in posting verbs. The dotted arrows are PCIe PIO operations. The solid, straight arrows are DMA operations: the thin ones are for writing the completion events. The thick wavy arrows are RDMA data packets and the thin ones are ACKs.

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 frame-

continued on page 26

continued from page 25

works with a lower time-to-science to post-process their huge datasets after execution, there is increasing pressure to enable use of MapReduce within these traditional High Performance Computing (HPC) architectures.

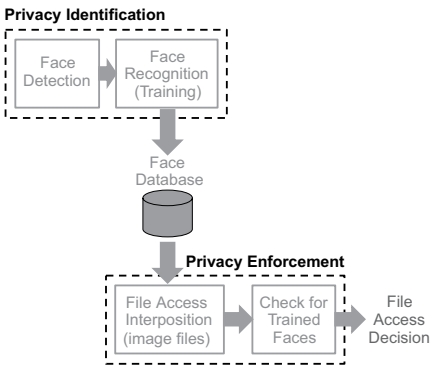
Therefore, in this work we explore potential means to enable use of the easy-to-program Hadoop MapReduce framework without requiring a complete infrastructure overhaul from existing HPC NAS solutions. We demonstrate that retaining function-dedicated resources like NAS is not only possible, but can even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

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 ne-grained runtime access control system for stored photos for Android which requires minimal user assistance, runs entirely



Overall approach of CHIPS.

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%.

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.

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

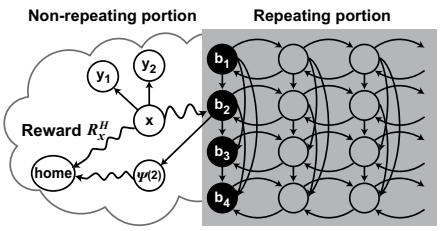
Anshul Gandhi, Sherwin Doroudi, Mor Harchol-Balter & Alan Scheller-Wolf

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-

continued on page 27

continued from page 27



This figure depicts the class of Markov chains that can be analyzed in *closed-form* via our renewal-based approach. In this class, the horizontal transitions are skip-free and the vertical transitions are unidirectional. The repeating portion is highlighted in *gray* and the border states, b_i, are shaded *black*. Note that y_i are the neighbors of x, and $\Psi(2)$ is the exit state in the non-repeating portion accessible from the border state b_2.

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.

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 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.

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.

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

continued on page 28

continued from page 27

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.

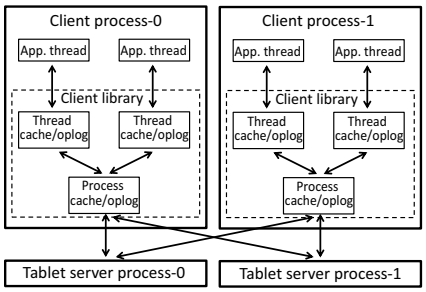
Our performance results demonstrate that our new hash table design—based around optimistic cuckoo hashing—outperforms other optimized concurrent hash tables by up to 2.5x for write-heavy workloads, even while using substantially less memory for small key-value items. On a 16-core machine, our hash table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

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.



LazyTable running two application processes with two application threads each.

2014 PDL Workshop and Retreat.