Parallel Data Laboratory
Carnegie Mellon University

FALL UPDATE

PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • FALL 2017

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

PDL CONSORTIUM MEMBERS

Broadcom, Ltd.
Citadel
Dell EMC
Google
Hewlett-Packard Labs
Hitachi, Ltd.
Intel Corporation
Microsoft Research
MongoDB
NetApp, Inc.
Oracle Corporation
Salesforce
Samsung Information Systems America
Seagate Technology
Two Sigma
Toshiba
Veritas
Western Digital

CONTENTS

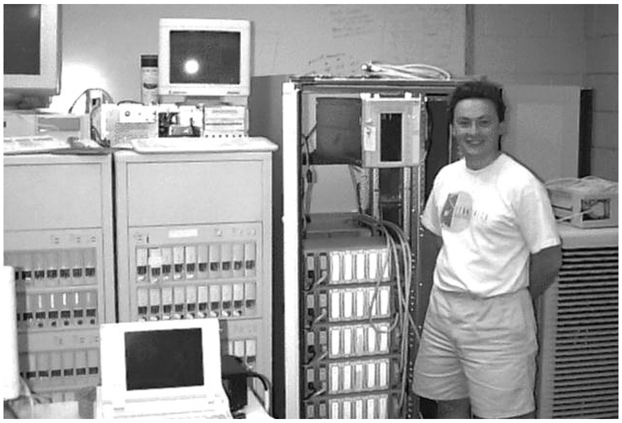
THE PDL PACKET

EDITOR
Joan Digney
CONTACTS
Greg Ganger
PDL Director
Bill Courtright
PDL Executive Director
Karen Lindenfelser
PDL Administrative Manager
The Parallel Data Laboratory
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891
TEL 412-268-6716
FAX 412-268-3010

The PDL is 25!

by Joan Digney, Greg Ganger & Bill Courtright

After a successful formative workshop in late 1992, Dr. Garth Gibson officially launched the PDL in 1993 with 7 students from CMU's CS and ECE Departments. Having recently finished his Ph.D. research, which defined the industry standard RAID terminology for redundant disk arrays, Gibson guided PDL researchers in advanced disk array research. The name "Parallel Data Lab" came from this initial focus on parallelism in storage systems. In the PDL's formative years, its researchers developed technologies for improving failure recovery performance (parity declustering) and maximizing performance in small-write intensive workloads (parity logging). They also developed an aggressive prefetching technology (transparent informed prefetching, or TIP) for converting serial access patterns into highly parallel workloads capable of exploiting large disk arrays.



Garth and the Scotch Parallel File System

The PDL first received actual lab space at CMU (Wean Hall 3607) to go with its name in January of 1994. As it grew, PDL became more spread out, with people in various areas of Wean Hall and the D-Level of Hamerschlag Hall. Today, PDL people are primarily located in the Robert Mehrabian Collaborative Innovation Center (RMCIC) and the Gates Center for Computer Science.The equipment populating PDL lab spaces is often state of the art, a rare case for academic research, and is upgraded frequently as technology advances. In large part, we have our sponsor companies to thank for this.

PDL's initial seed funding came from CMU's Data Storage Systems Center (DSSC), then directed by Mark Kryder, and from DARPA (from which much PDL funding has come over the years). Additional original funding came from the member companies of the PDL Consortium, whose initial members were AT&T Global Information Systems, Data General, IBM, Hewlett-Packard, Seagate and Storage Technology. Today, PDL funding comes from DoD, DoE, NSF, and the current PDL Consortium members. The list of consortium members has become long over the years, as companies join, merge, and change research focus. It is currently comprised of the following: Broadcom, Ltd., Citadel, Dell EMC, Google, Hewlett-Packard Labs, Hitachi, Ltd., Intel Corporation, Microsoft Research, MongoDB,

continued on page 2

continued from page 1

NetApp, Inc., Oracle Corporation, Salesforce, Samsung Information Systems America, Seagate Technology, Two Sigma, Toshiba, Veritas, and Western Digital. VMware has also been generous with their financial support.

In 1995, Gibson and Dr. David Nagle launched a new PDL project called Network-Attached Secure Disks (NASD), a network-attached storage architecture for achieving cost-effective scalable bandwidth. In addition to their research advances, Gibson founded and chaired an industry working group within the Information Storage Industry Consortium (INSIC) to transfer the new technology and move towards standardization of the NASD architecture.

In 1999, Nagle took over as PDL Director when Gibson went on leave to co-found Panasas. In 2000, Dr. Greg Ganger, who joined the ECE faculty and the PDL in 1997, jointly directed the PDL with Nagle, then became Director in 2001 when Nagle went on leave. Early research initiatives under Ganger's leadership included Self-Securing Devices, PASIS (Survivable Storage), and Self-* (tuning, managing, …) Storage.

In 2006, after several years of preparation, the PDL opened the Data Center Observatory (DCO), a machine room with over 1,875 square feet of space. As of May 2017, it is populated with 1077 computers, connected to 76 switches, 111 power distributors and 22 remote console servers with a total of 3,683 cables. The DCO provides a computation and storage utility to resource-hungry research activities such as data mining, design simulation, network intrusion detection, and visualization. The DCO houses several large research clusters including Susitna, Marmot and Narwhal.

An early PDL group photo, ca. ~1995.

Garth and students; from L to R, Dan Stodolsky, Hugo Patterson, Garth, and Bill Courtright.

Since then, each year has seen the development of many exiting ideas in the PDL. Here are a few examples:

» DISC (data-intensive supercomputing) allowed applications to extract deep insight from huge and dynamically-changing datasets.

» We explored the use of virtual machines using FSVAs (file system virtual appliances) to address the porting problems associated with the client-side component of most cluster-based designs (including ours).

» The Perspective home storage system was designed to simplify data management and sharing among the many storage-enhanced devices (e.g., DVRs, iPODs, laptops).

» Led by Prof. David Andersen, the FAWN (fast array of wimpy nodes) project explored new cluster architectures able to provide data-intensive computing with order of magnitude improvements in energy efficiency.

» We continue to research the technology advances needed for cloud computing. Our OpenCloud and OpenCirrus clouds provide resources for real users, as well as provide us with invaluable Hadoop logs, instrumentation data, and case studies.

» We have explored exciting new storage technologies, such as NVM and Flash SSDs. Even the disk drive is changing, with technologies like shingled magnetic recording creating a need to reconsider usage patterns and interfaces.

» Focus has reemerged on database systems, including work on automated database tuning, deduplication in databases, incremental computation, and more exploitation of NVM in databases. In particular, Andy Pavlo's Peloton DBMS project combines several of these activities into what he refers to as a "self-driving" database, seeking to achieve autonomous adaptation to workload and resource conditions.

» The breadth of analytics frameworks and other cloud computing activities continues to grow and has led to resource scheduling challenges. Our TetriSched project developed new ways of allowing users to express their per-job resource type preferences (e.g., machine locality or hardware accelerators) and explored the trade-

continued on page 16

October 2017
Lorrie Cranor Awarded FORE Systems Chair of Computer Science

We are very pleased to announce that, in addition to a long list of accomplishments, which has included a term as the Chief Technologist of the Federal Trade Commission, Lorrie Cranor has been made the FORE Systems Professor of Computer Science and Engineering & Public Policy at CMU.

Lorrie provided information that "the founders of FORE Systems, Inc. established the FORE Systems Professorship in 1995 to support a faculty member in the School of Computer Science. The company's name is an acronym formed by the initials of the founders' first names. Before it was acquired by Great Britain's Marconi in 1998, FORE created technology that allows computer networks to link and transfer information at a rapid speed. Ericsson purchased much of Marconi in 2006." The chair was previously held by CMU University Professor Emeritus, Edmund M. Clarke.

September 2017
Garth Gibson to Lead New Vector Institute for AI in Toronto

In January of 2018, PDL's founder, Garth Gibson, will become President and CEO of the Vector Institute for AI in Toronto. Vector's website states that "Vector will be a leader in the transformative field of artificial intelligence, excelling in machine and deep learning — an area of scientific, academic, and commercial endeavour that will shape our world over the next generation."

Frank Pfenning, Head of the Department of Computer Science, notes that "this is a tremendous opportunity for Garth, but we will sorely miss him in the multiple roles he plays in the department and school: Professor (and all that this entails), Co-Director of the MCDS program, and Associate Dean for Masters Programs in SCS."

We are sad to see him go and will miss him greatly, but the opportunities presented here for world level innovation are tremendous and we wish him all the best.

June 2017
Dana Van Aken's SIGMOD Paper Featured in Amazon AI Blog

Please see Amazon's AI blog at https://aws.amazon.com/blogs/ai/tuning-your-dbms-automatically-with-machine-learning/ to read about Tuning Your DBMS Automatically with Machine Learning, an article by Dana Van Aken, based on the SIGMOD '17 paper "Automatic Database Management System Tuning Through Large-scale Machine Learning," which she co-authored with Andy Pavlo and Geoff Gordon.

June 2017
Satya and Colleagues Honored for Creation of Andrew File System

The Association for Computing Machinery has named the developers of Carnegie Mellon University's pioneering Andrew File System (AFS) the recipients of its prestigious 2016 Software System Award.

AFS was the first distributed file system designed for tens of thousands of machines, and pioneered the use of scalable, secure and ubiquitous access to shared file data. To achieve the goal of providing a common shared file system used by large networks of people, AFS introduced novel approaches to caching, security, management and administration.

The award recipients, including Computer Science Professor Mahadev Satyanarayanan, built the Andrew File System in the 1980s while working as a team at the Information Technology Center (ITC) — a partnership between Carnegie Mellon and IBM.

The ACM Software System Award is presented to an institution or individuals recognized for developing a software system that has had a lasting influence, reflected in contributions to concepts, in commercial acceptance, or both.

AFS is still in use today as both an open-source system and as a file system in commercial applications. It has also inspired several cloud-based storage applications. Many universities integrated AFS before it was introduced as a commercial application.

In addition to Satya, the recipients of the award include former faculty member Alfred Z. Spector, alumnus Michael L. Kazar, Robert N. Sidebotham, David A. Nichols, Michael J. West, John H. Howard and Sherri M. Nichols. Many of them also contributed to two foundational AFS papers: "The ITC Distributed File System: Principles and Design," published in Proceedings of ACM SOSP 1985, and "Scale and Performance in a Distributed File System," published in Proceedings of ACM SOSP 1987. The Software System Award carries a prize of $35,000. Financial support for the Software System Award is provided by IBM.

-- Byron Spice, The Piper, June 1, 2017

DISSERTATION ABSTRACT: Understanding and Improving the Latency of DRAM-Based Memory System

Kevin K. Chang
Carnegie Mellon University, ECE

PhD Defense — May 5, 2017

Over the past two decades, the storage capacity and access bandwidth of main memory have improved tremendously, by 128x and 20x, respectively. These improvements are mainly due to the continuous technology scaling of DRAM (dynamic random-access memory), which has been used as the physical substrate for main memory. In stark contrast with capacity and bandwidth, DRAM latency has remained almost constant, reducing by only 1.3x in the same time frame. Therefore, long DRAM latency continues to be a critical performance bottleneck in modern systems. Increasing core counts, and the emergence of increasingly more data-intensive and latency-critical applications further stress the importance of providing low-latency memory accesses.

In this dissertation, we identify three main problems that contribute significantly to long latency of DRAM accesses. To address these problems, we present a series of new techniques. Our new techniques significantly improve both system performance and energy efficiency. We also examine the critical relationship between supply voltage and latency in modern DRAM chips and develop new mechanisms that exploit this voltage-latency tradeoff to improve energy efficiency.

First, while bulk data movement is a key operation in many applications and operating systems, contemporary systems perform this movement inefficiently, by transferring data from DRAM to the processor, and then back to DRAM, across a narrow off-chip channel. The use of this narrow channel for bulk data movement results in high latency and high energy consumption. This dissertation introduces a new DRAM design, Low-cost Inter-linked SubArrays (LISA), which provides fast and energy-efficient bulk data movement across sub- arrays in a DRAM chip. We show that the LISA substrate is very powerful and versatile by demonstrating that it efficiently enables several new architectural mechanisms, including low-latency data copying, reduced DRAM access latency for frequently-accessed data, and reduced preparation latency for subsequent accesses to a DRAM bank.

Thomas Kim presents a demo at the 2017 PDL Spring Visit Day.

Second, DRAM needs to be periodically refreshed to prevent data loss due to leakage. Unfortunately, while DRAM is being refreshed, a part of it becomes unavailable to serve memory requests, which degrades system performance. To address this refresh interference problem, we propose two access-refresh parallelization techniques that enable more overlap- ping of accesses with refreshes inside DRAM, at the cost of very modest changes to the memory controllers and DRAM chips. These two techniques together achieve performance close to an idealized system that does not require refresh.

Third, we find, for the first time, that there is significant latency variation in accessing different cells of a single DRAM chip due to the irregularity in the DRAM manufacturing process. As a result, some DRAM cells are inherently faster to access, while others are inherently slower. Unfortunately, existing systems do not exploit this variation and use a fixed latency value based on the slowest cell across all DRAM chips. To exploit latency variation within the DRAM chip, we experimentally characterize and understand the behavior of the variation that exists in real commodity DRAM chips. Based on our characterization, we propose Flexible-LatencY DRAM (FLY-DRAM), a mechanism to reduce DRAM latency by categorizing the DRAM cells into fast and slow regions, and accessing the fast regions with a reduced latency, thereby improving system performance significantly. Our extensive experimental characterization and analysis of latency variation in DRAM chips can also enable development of other new techniques to improve performance or reliability.

Fourth, this dissertation, for the first time, develops an understanding of the latency behavior due to another important factor—supply voltage, which significantly impacts DRAM performance, energy consumption, and reliability. We take an experimental approach to understanding and exploiting the behavior of modern DRAM chips under different supply voltage values. Our detailed characterization of real commodity DRAM chips demonstrates that memory access latency reduces with increasing supply voltage. Based on our characterization, we propose Voltron, a new mechanism that improves system energy efficiency by dynamically adjusting the DRAM supply voltage based on a performance model. Our extensive experimental data on the relationship between DRAM supply voltage, latency, and reliability can further enable developments of other new mechanisms that improve latency, energy efficiency, or reliability.

The key conclusion of this dissertation is that augmenting DRAM architecture with simple and low-cost features, and developing a better understanding of manufactured DRAM chips together leads to significant memory latency reduction as well as energy efficiency improvement. We hope and believe that the proposed architectural techniques and

continued on page 5

continued from page 4

detailed experimental data on real commodity DRAM chips presented in this dissertation will enable developments of other new mechanisms to improve the performance, energy efficiency, or reliability of future memory systems.

DISSERTATION ABSTRACT: Meeting Tail Latency SLOs in Shared Networked Storage

Timothy Zhu
Carnegie Mellon University, SCS

PhD Defense — May 3, 2017

Shared computing infrastructures (e.g., cloud computing, enterprise datacenters) have become the norm today due to their lower operational costs and IT management costs. However, resource sharing introduces challenges in controlling performance for each of the workloads using the infrastructure. For user-facing workloads (e.g., web server, email server), one of the most important performance metrics companies want to control is tail latency, the time it takes to complete the most delayed requests. Ideally, companies would be able to specify tail latency performance goals, also called Service Level Objectives (SLOs), to ensure that almost all requests complete quickly.

Meeting tail latency SLOs is challenging for multiple reasons. First, tail latency is significantly affected by the burstiness that is commonly exhibited by production workloads. Burstiness leads to transient queueing, which is a major cause of high tail latency. Second, tail latency is often due to I/O (e.g., storage, networks), and I/O devices exhibit performance peculiarities that make it hard to meet SLOs. Third, the end-to-end latency is affected by sum of latencies across multiple types of resources such as storage and networks. Most of the existing research, however, have ignored burstiness and focused on a single resource.

This thesis introduces new techniques for meeting end-to-end tail latency SLOs in both storage and networks while accounting for the burstiness that arises in production workloads. We address open questions in scheduling policies, admission control, and workload placement. We build a new Quality of Service (QoS) system for meeting tail latency SLOs in networked storage infrastructures. Our system uses prioritization and rate limiting as tools for controlling the congestion between workloads. We introduce a novel approach for intelligently configuring the workload priorities and rate limits using two different types of queueing analyses: Deterministic Network Calculus (DNC) and Stochastic Network Calculus (SNC). By integrating these mathematical analyses into our system, we are able to build better algorithms for optimizing the resource usage. Our implementation results using realistic workload traces on a physical cluster demonstrate that our approach can meet tail latency SLOs while achieving better resource utilization than the state-of-the-art.

While this thesis focuses on scheduling policies, admission control, and workload placement in storage and networks, the ideas presented in our work can be applied to other related problems such as workload migration and datacenter provisioning. Our theoretically grounded techniques for controlling tail latency can also be extended beyond storage and networks to other contexts such as the CPU, cache, etc. For example, in real-time CPU scheduling contexts, our DNC-based techniques could be used to provide strict latency guarantees while accounting for workload burstiness.

Aaron Harlap talks about his research at the 2016 PDL retreat.

THESIS PROPOSAL: The Design & Implementation of a Non-Volatile Memory Database Management System

Joy Arulraj, SCS

October 20, 2017

For the first time in 25 years, a new non-volatile memory (NVM) category is being created that is two orders of magnitude faster than current durable storage media. This will fundamentally change the dichotomy between volatile memory and durable storage in DB systems. The new NVM devices are almost as fast as DRAM, but all writes to it are potentially persistent even after power loss. Existing DB systems 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 components of legacy database systems are unnecessary and will degrade the performance of data intensive applications.

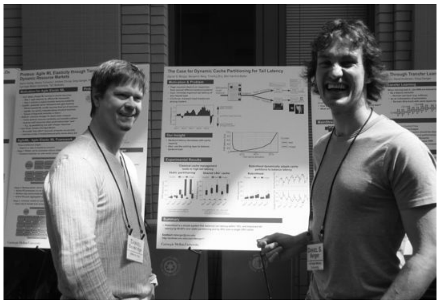
This dissertation explores the implications of NVM for database systems. It presents the design and implementation of Peloton, a new database system tailored specifically for NVM. We focus on three aspects of a database system: (1) logging and recovery, (2) storage management, and (3) indexing. Our primary contribution in this dissertation is the design of a new logging and recovery protocol, called write-behind logging, that improves the availability of the system by more than two orders of magnitude compared to the ubiquitous write- ahead logging protocol. Besides improving availability, we found that write- behind logging improves the space utilization of the NVM device and extends its lifetime. Second, we propose a new storage engine architecture that leverages the durability and byte-addressability properties of NVM to avoid unnecessary data duplication. Third, the dissertation presents the design of a latch-free range index tailored for NVM that supports near-instantaneous recovery without requiring special-purpose recovery code.

Software-Defined Storage for Fast Trajectory Queries using a DeltaFS Indexed Massive Directory

Qing Zheng, George Amvrosiadis, Saurabh Kadekodi, Garth Gibson, Chuck Cranor, Brad Settlemyer, Gary Grider & Fan Guo

PDSW-DISCS 2017: 2nd Joint International Workshop on Parallel Data Storage and Data Intensive Scalable Computing System held in conjunction with SC17, Denver, CO, Nov. 2017.

In this paper we introduce the Indexed Massive Directory, a new technique for indexing data within DeltaFS. With its design as a scalable, server-less file system for HPC platforms, DeltaFS scales file system metadata performance with application scale. The Indexed Massive Directory is a novel extension to the DeltaFS data plane, enabling in-situ indexing of massive amounts of data written to a single directory simultaneously, and in an arbitrarily large number of files. We achieve this through a memory-efficient indexing mechanism for reordering and indexing writes, and a log-structured storage layout to pack small data into large log objects, all while ensuring compute node resources are used frugally. We demonstrate the efficiency of this indexing mechanism through VPIC, a plasma simulation code that scales to trillions of particles. With Indexed Massive Directory, we modify VPIC to create a file for each particle to receive writes of that particle's simulation output data. Dynamically indexing the directory's underlying storage keyed on particle filename allows us to achieve a 5000x speedup for a single particle trajectory query, which requires reading all data for a single particle. This speedup increases with application scale, while the overhead remains stable at 3% of the available memory.



Daniel Beveridge, of VMware, Inc. discusses research with Daniel Berger at the 2017 PDL Spring Visit Day.

Bigger, Longer, Fewer: What Do Cluster Jobs Look Like Outside Google?

George Amvrosiadis, Jun Woo Park, Gregory R. Ganger, Garth A. Gibson, Elisabeth Baseman & Nathan DeBardeleben

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-17-104, October 2017.

In the last 5 years, a set of job scheduler logs released by Google has been used in more than 400 publications as the token cloud workload. While this is an invaluable trace, we think it is crucial that researchers evaluate their work under other workloads as well, to ensure the generality of their techniques. To aid them in this process, we analyze three new traces consisting of job scheduler logs from one private and two HPC clusters. We further release the two HPC traces, which we expect to be of interest to the community due to their unique characteristics. The new traces represent clusters 0.3-3 times the size of the Google cluster in terms of CPU cores, and cover a 3-60 times longer time span.

This paper presents an analysis of the differences and similarities between all aforementioned traces. We discuss a variety of aspects: job characteristics, workload heterogeneity, resource utilization, and failure rates. More importantly, we review assumptions from the literature that were originally derived from the Google trace, and verify whether they hold true when the new traces are considered. For those assumptions that are violated, we examine affected work from the literature. Finally, we demonstrate the importance of dataset plurality in job scheduling research by evaluating the performance of JVuPredict, the job runtime estimate module of the TetriSched scheduler, using all four traces.

Mosaic: A GPU Memory Manager with Application-Transparent Support for Multiple Page Sizes

Rachata Ausavarungnirun, Joshua Landgraf, Vance Miller, Saugata Ghose, Jayneel Gandhi, Christopher J. Rossbach & Onur Mutlu

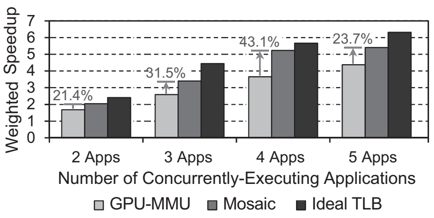
Proc. of the International Symposium on Microarchitecture (MICRO), Cambridge, MA, October 2017.

Contemporary discrete GPUs support rich memory management features such as virtual memory and demand paging. These features simplify GPU programming by providing a virtual address space abstraction similar to CPUs and eliminating manual memory management, but they introduce high performance overheads during (1) address translation and (2) page faults. A GPU relies on high degrees of thread-level parallelism (TLP) to hide memory latency. Address translation can undermine TLP, as a single miss in the translation lookaside buffer (TLB) invokes an expensive serialized page table walk that often stalls multiple threads. Demand paging can also undermine TLP, as multiple threads often stall while they wait for an expensive data transfer over the system I/O (e.g., PCIe) bus when the GPU demands a page.

In modern GPUs, we face a tradeoff on how the page size used for memory management affects address translation and demand paging. The address translation overhead is lower when we employ a larger page size (e.g., 2MB large pages, compared

continued on page 7

continued from page 6



Heterogeneous workload performance of the GPU memory managers.

with conventional 4KB base pages), which increases TLB coverage and thus reduces TLB misses. Conversely, the demand paging overhead is lower when we employ a smaller page size, which decreases the system I/O bus transfer latency. Support for multiple page sizes can help relax the page size trade-off so that address translation and demand paging optimizations work together synergistically. However, existing page coalescing (i.e., merging base pages into a large page) and splintering (i.e., splitting a large page into base pages) policies require costly base page migrations that undermine the benefits multiple page sizes provide. In this paper, we observe that GPGPU applications present an opportunity to support multiple page sizes without costly data migration, as the applications perform most of their memory allocation en masse (i.e., they allocate a large number of base pages at once). We show that this en masse allocation allows us to create intelligent memory allocation policies which ensure that base pages that are contiguous in virtual memory are allocated to contiguous physical memory pages. As a result, coalescing and splintering operations no longer need to migrate base pages.

We introduce Mosaic, a GPU memory manager that provides application-transparent support for multiple page sizes. Mosaic uses base pages to transfer data over the system I/O bus, and allocates physical memory in a way that (1) preserves base page contiguity and (2) ensures that a large page frame contains pages from only a single memory protection domain. We take advantage of this allocation strategy to design a novel in-place page size selection mechanism that avoids data migration. This mechanism allows the TLB to use large pages, reducing address translation overhead. During data transfer, this mechanism enables the GPU to transfer only the base pages that are needed by the application over the system I/O bus, keeping demand paging overhead low. Our evaluations show that Mosaic reduces address translation overheads while efficiently achieving the benefits of demand paging, compared to a contemporary GPU that uses only a 4KB page size. Relative to a state-of-the-art GPU memory manager, Mosaic improves the performance of homogeneous and heterogeneous multi-application workloads by 55.5% and 29.7% on average, respectively, coming within 6.8% and 15.4% of the performance of an ideal TLB where all TLB requests are hits.

Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology

Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons & Todd C. Mowry

Proceedings of the 50th International Symposium on Microarchitecture (MICRO), Boston, MA, USA, October 2017.

Many important applications trigger bulk bitwise operations, i.e., bitwise operations on large bit vectors. In fact, recent works design techniques that exploit fast bulk bitwise operations to accelerate databases (bitmap indices, BitWeaving) and web search (BitFunnel). Unfortunately, in existing architectures, the throughput of bulk bitwise operations is limited by the memory bandwidth available to the processing unit (e.g., CPU, GPU, FPGA, processing-in-memory). To overcome this bottleneck, we propose Ambit, an Accelerator-in-Memory for bulk bitwise operations. Unlike prior works, Ambit exploits the analog operation of DRAM technology to perform bitwise operations completely inside DRAM, thereby exploiting the full internal DRAM bandwidth.

Ambit consists of two components. First, simultaneous activation of three DRAM rows that share the same set of sense amplifiers enables the system to perform bitwise AND and OR operations. Second, with modest changes to the sense amplifier, the system can use the inverters present inside the sense amplifier to perform bitwise NOT operations. With these two components, Ambit can perform any bulk bitwise operation efficiently inside DRAM. Ambit largely exploits existing DRAM structure, and hence incurs low cost on top of commodity DRAM designs (1% of DRAM chip area). Importantly, Ambit uses the modern DRAM interface without any changes, and therefore it can be directly plugged onto the memory bus. Our extensive circuit simulations show that Ambit works as expected even in the presence of significant process variation.

Averaged across seven bulk bitwise operations, Ambit improves performance by 32X and reduces energy consumption by 35X compared to state-of-the-art systems. When integrated with Hybrid Memory Cube (HMC), a 3D-stacked DRAM with a logic layer, Ambit improves performance of bulk bitwise operations by 9.7X compared to processing in the logic layer of the HMC. Ambit improves the performance of three real-world data-intensive applications, 1) database bitmap indices, 2) BitWeaving, a technique to accelerate database scans, and 3) bit-vector-based implementation of sets, by 3X-7X compared to a state-of-the-art baseline using SIMD optimizations. We describe four other applications that can benefit from Ambit, including a recent technique

continued on page 8

continued from page 7

proposed to speed up web search. We believe that large performance and energy improvements provided by Ambit can enable other applications to use bulk bitwise operations.

Detecting and Mitigating Data-Dependent DRAM Failures by Exploiting Current Memory Content

Samira Khan, Chris Wilkerson, Zhe Wang, Alaa R. Alameldeen, Donghyuk Lee & Onur Mutlu

Proceedings of the 50th International Symposium on Microarchitecture (MICRO), Boston, MA, USA, October 2017.

DRAM cells in close proximity can fail depending on the data content in neighboring cells. These failures are called data-dependent failures. Detecting and mitigating these failures online, while the system is running in the field, enables various optimizations that improve reliability, latency, and energy efficiency of the system. For example, a system can improve performance and energy efficiency by using a lower refresh rate for most cells and mitigate the failing cells using higher refresh rates or error correcting codes. All these system optimizations depend on accurately detecting every possible data-dependent failure that could occur with any content in DRAM. Unfortunately, detecting all data-dependent failures requires the knowledge of DRAM internals specific to each DRAM chip. As internal DRAM architecture is not exposed to the system, detecting data-dependent failures at the system-level is a major challenge.

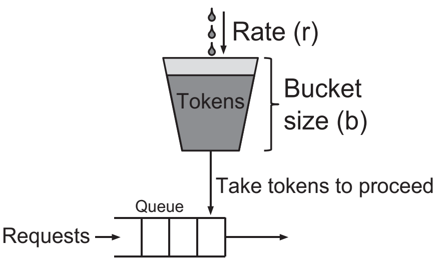
In this paper, we decouple the detection and mitigation of data-dependent failures from physical DRAM organization such that it is possible to detect failures without knowledge of DRAM internals. To this end, we propose MEMCON, a memory content-based detection and mitigation mechanism for data-dependent failures in DRAM. MEMCON does not detect every possible data-dependent failure. Instead, it detects and mitigates failures that occur only with the current content in memory while the programs are running in the system. Such a mechanism needs to detect failures whenever there is a write access that changes the content of memory. As detection of failure with a runtime testing has a high overhead, MEMCON selectively initiates a test on a write, only when the time between two consecutive writes to that page (i.e., write interval) is long enough to provide significant benefit by lowering the refresh rate during that interval. MEMCON builds upon a simple, practical mechanism that predicts the long write intervals based on our observation that the write intervals in real workloads follow a Pareto distribution: the longer a page remains idle after a write, the longer it is expected to remain idle. Our evaluation shows that compared to a system that uses an aggressive refresh rate, MEMCON reduces refresh operations by 65-74%, leading to a 10%/17%/40% (min) to 12%/22%/50% (max) performance improvement for a single-core and 10%/23%/52% (min) to 17%/29%/65% (max) performance improvement for a 4-core system using 8/16/32 Gb DRAM chips.

Workload Compactor: Reducing Datacenter Cost while ProvidingTail Latency SLO Guarantees

Timothy Zhu, Michael A. Kozuch & Mor Harchol-Balter

ACM Symposium on Cloud Computing (SoCC'17) , Santa Clara, Oct 2017.

Service providers want to reduce datacenter costs by consolidating workloads onto fewer servers. At the same time, customers have performance goals, such as meeting tail latency Service Level Objectives (SLOs). Consolidating workloads while meeting tail latency goals is challenging, especially since workloads in production environments are often bursty. To limit the congestion when consolidating workloads, customers and service providers often agree upon rate limits. Ideally, rate limits are chosen to maximize the number of workloads that can be co-located while meeting each workload's SLO. In reality, neither the service provider nor customer knows how to choose rate limits. Customers end up selecting rate limits on their own in some ad hoc fashion, and service providers are left to optimize given the chosen rate limits.



Token bucket rate limiters control the rate and burstiness of a stream of requests. When a request arrives at the rate limiter, tokens are used (i.e., removed) from the token bucket to allow the request to proceed. If the bucket is empty, the request must queue and wait until there are enough tokens. Tokens are added to the bucket at a constant rate r up to a maximum capacity as specified by the bucket size b. Thus, the token bucket rate limiter limits the workload to a maximum instantaneous burst of size b and an average rate r.

This paper describes WorkloadCompactor, a new system that uses workload traces to automatically choose rate limits simultaneously with selecting onto which server to place workloads. Our system meets customer tail latency SLOs while minimizing datacenter resource costs. Our experiments show that by optimizing the choice of rate limits, WorkloadCompactor reduces the number of required servers by 30-60% as compared to state-of-the-art approaches.

Error Characterization,

continued on page 9

continued from page 8

Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives

Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo & Onur Mutlu

Proceedings of the IEEE Volume: 105, Issue: 9, Sept. 2017.

NAND flash memory is ubiquitous in everyday life today because its capacity has continuously increased and cost has continuously decreased over decades. This positive growth is a result of two key trends: 1) effective process technology scaling; and 2) multi-level (e.g., MLC, TLC) cell data coding. Unfortunately, the reliability of raw data stored in flash memory has also continued to become more difficult to ensure, because these two trends lead to 1) fewer electrons in the flash memory cell floating gate to represent the data; and 2) larger cell-to-cell interference and disturbance effects. Without mitigation, worsening reliability can reduce the lifetime of NAND flash memory. As a result, flash memory controllers in solid-state drives (SSDs) have become much more sophisticated: they incorporate many effective techniques to ensure the correct interpretation of noisy data stored in flash memory cells. In this article, we review recent advances in SSD error characterization, mitigation, and data recovery techniques for reliability and lifetime improvement. We provide rigorous experimental data from state-of-the-art MLC and TLC NAND flash devices on various types of flash memory errors, to motivate the need for such techniques. Based on the understanding developed by the experimental characterization, we describe several mitigation and recovery techniques, including 1) cell-to-cell interference mitigation; 2) optimal multi-level cell sensing; 3) error correction using state-of-the-art algorithms and methods; and 4) data recovery when error correction fails. We quantify the reliability improvement provided by each of these techniques. Looking forward, we briefly discuss how flash memory and these techniques could evolve into the future.

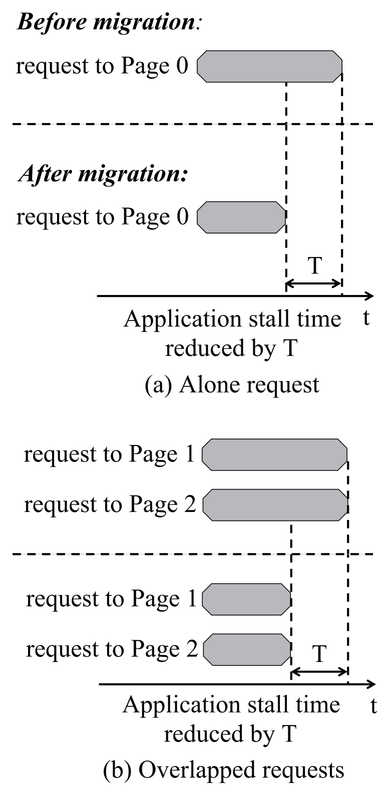
Utility-Based Hybrid Memory Management

Yang Li, Saugata Ghose, Jongmoo Choi, Jin Sun, Hui Wang & Onur Mutlu

In Proc. of the IEEE Cluster Conference (CLUSTER), Honolulu, HI, September 2017.

While the memory footprints of cloud and HPC applications continue to increase, fundamental issues with DRAM scaling are likely to prevent traditional main memory systems, composed of monolithic DRAM, from greatly growing in capacity. Hybrid memory systems can mitigate the scaling limitations of monolithic DRAM by pairing together multiple memory technologies (e.g., different types of DRAM, or DRAM and non-volatile memory) at the same level of the memory hierarchy. The goal of a hybrid main memory is to combine the different advantages of the multiple memory types in a cost-effective manner while avoiding the disadvantages of each technology. Memory pages are placed in and migrated between the different memories within a hybrid memory system, based on the properties of each page. It is important to make intelligent page management (i.e., placement and migration) decisions, as they can significantly affect system performance.

In this paper, we propose utility-based hybrid memory management (UH-MEM), a new page management mechanism for various hybrid memories, that systematically estimates the utility (i.e., the system performance benefit) of migrating a page between different memory types, and uses this information to guide data placement. UH-MEM operates in two steps. First, it estimates how much a single application would benefit from migrating one of its pages to a different type of memory, by comprehensively considering access frequency, row buffer locality, and memory-level parallelism. Second, it translates the estimated benefit of a single application to an estimate of the overall system performance benefit from such a migration.



Conceptual example showing that the MLP of a page influences how much effect its migration to fast memory has on the application stall time.

We evaluate the effectiveness of UH-MEM with various types of hybrid memories, and show that it significantly improves system performance on each of these hybrid memories. For a memory system with DRAM and non-volatile memory, UH-MEM improves performance by 14% on average (and up to 26%) compared to the best of three evaluated state-of-the-art mechanisms across a large number of data-intensive workloads.

continued on page 10

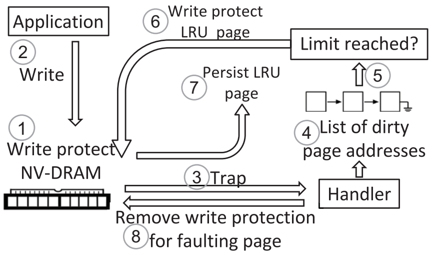
continued from page 9

Viyojit: Decoupling Battery and DRAM Capacities for Battery-Backed DRAM.

Rajat Kateja, Anirudh Badam, Sriram Govindan, Bikash Sharma & Greg Ganger

ISCA '17, June 24-28, 2017, Toronto, ON, Canada.

Non-Volatile Memories (NVMs) can significantly improve the performance of data-intensive applications. A popular form of NVM is Battery-backed DRAM, which is available and in use today with DRAMs latency and without the endurance problems of emerging NVM technologies. Modern servers can be provisioned with up-to 4 TB of DRAM, and provisioning battery backup to write out such large memories is hard because of the large battery sizes and the added hardware and cooling costs. We present Viyojit, a system that exploits the skew in write working sets of applications to provision substantially smaller batteries while still ensuring durability for the entire DRAM capacity. Viyojit achieves this by bounding the number of dirty pages in DRAM based on the provisioned battery capacity and proactively writing out infrequently written pages to an SSD. Even for write-heavy workloads with less skew than we observe in analysis of real data center traces, Viyojit reduces the required battery capacity to 11% of the original size, with a performance overhead of 7-25%. Thus, Viyojit frees battery-backed DRAM from stunted growth of battery capacities and enables servers with terabytes of battery-backed DRAM.



Flow chart describing Viyojit's implementation for tracking dirty pages and enforcing the dirty budget.

Scheduling for Efficiency and Fairness in Systems with Redundancy

Kristen Gardner, Mor Harchol-Balter, Esa Hyyti & Rhonda Righter

Performance Evaluation, July 2017.

Server-side variability—the idea that the same job can take longer to run on one server than another due to server-dependent factors—is an increasingly important concern in many queueing systems. One strategy for overcoming server-side variability to achieve low response time is redundancy, under which jobs create copies of themselves and send these copies to multiple different servers, waiting for only one copy to complete service. Most of the existing theoretical work on redundancy has focused on developing bounds, approximations, and exact analysis to study the response time gains offered by redundancy. However, response time is not the only important metric in redundancy systems: in addition to providing low overall response time, the system should also be fair in the sense that no job class should have a worse mean response time in the system with redundancy than it did in the system before redundancy is allowed.

In this paper we use scheduling to address the simultaneous goals of (1) achieving low response time and (2) maintaining fairness across job classes. We develop new exact analysis for per-class response time under First-Come First-Served (FCFS) scheduling for a general type of system structure; our analysis shows that FCFS can be unfair in that it can hurt non-redundant jobs. We then introduce the Least Redundant First (LRF) scheduling policy, which we prove is optimal with respect to overall system response time, but which can be unfair in that it can hurt the jobs that become redundant. Finally, we introduce the Primaries First (PF) scheduling policy, which is provably fair and also achieves excellent overall mean response time.

A Better Model for Job Redundancy: Decoupling Server Slowdown and Job Size

Kristen Gardner, Mor Harchol-Balter, Alan Scheller-Wolf & Benny Van Houdt

Transactions on Networking, September 2017.

Recent computer systems research has proposed using redundant requests to reduce latency. The idea is to replicate a request so that it joins the queue at multiple servers. The request is considered complete as soon as any one of its copies completes. Redundancy allows us to overcome server-side variability – the fact that a server might be temporarily slow due to factors such as background load, network interrupts, and garbage collection – to reduce response time. In the past few years, queueing theorists have begun to study redundancy, first via approximations, and, more recently, via exact analysis. Unfortunately, for analytical tractability, most existing theoretical analysis has assumed an Independent Runtimes (IR) model, wherein the replicas of a job each experience independent runtimes (service times) at different servers. The IR model is unrealistic and has led to theoretical results which can be at odds with computer systems implementation results. This paper introduces a much more realistic model of redundancy. Our model decouples the inherent job size (X) from the server-side slowdown (S), where we track both S and X for each job. Analysis within the S&X model is, of course, much more difficult. Nevertheless, we design a dispatching policy, Redundant-to-Idle-Queue (RIQ), which is both analytically tractable within the S&X model and has provably excellent performance.

continued on page 11

continued from page 10

Litz: An Elastic Framework for High-Performance Distributed Machine Learning

Aurick Qiao, Abutalib Aghayev, Weiren Yu, Haoyang Chen, Qirong Ho, Garth A. Gibson & Eric P. Xing

Carnegie Mellon University Parallel Data Laboratory Technical Report CMU-PDL-17-103. June 2017.

Machine Learning (ML) is becoming an increasingly popular application in the cloud and data-centers, inspiring a growing number of distributed frameworks optimized for it. These frameworks leverage the specific properties of ML algorithms to achieve orders of magnitude performance improvements over generic data processing frameworks like Hadoop or Spark. However, they also tend to be static, unable to elastically adapt to the changing resource availability that is characteristic of the multi-tenant environments in which they run. Furthermore, the programming models provided by these frameworks tend to be restrictive, narrowing their applicability even within the sphere of ML workloads.

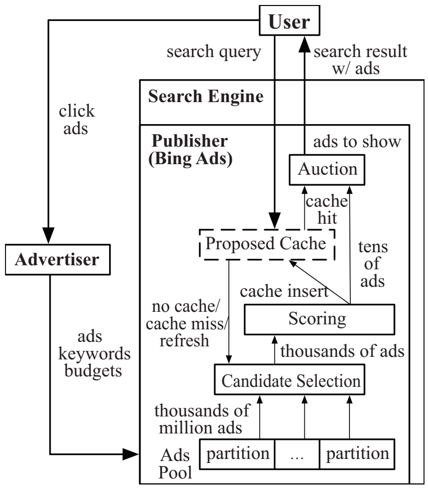
Motivated by these trends, we present Litz, a distributed ML framework that achieves both elasticity and generality without giving up the performance of more specialized frameworks. Litz uses a programming model based on scheduling micro-tasks with parameter server access which enables applications to implement key distributed ML techniques that have recently been introduced. Furthermore, we believe that the union of ML and elasticity presents new opportunities for job scheduling due to dynamic resource usage of ML algorithms. We give examples of ML properties which give rise to such resource usage patterns and suggest ways to exploit them to improve resource utilization in multi-tenant environments. To evaluate Litz, we implement two popular ML applications that vary dramatically terms of their structure and run-time behavior—they are typically implemented by different ML frameworks tuned for each. We show that Litz achieves competitive performance with the state of the art while providing low-overhead elasticity and exposing the underlying dynamic resource usage of ML applications.

Workload Analysis and Caching Strategies for Search Advertising Systems

Conglong Li, David G. Andersen, Qiang Fu, Sameh Elnikety & Yuxiong He

SoCC '17, September 24–27, 2017, Santa Clara, CA, USA..

Search advertising depends on accurate predictions of user behavior and interest, accomplished today using complex and computationally expensive machine learning algorithms that estimate the potential revenue gain of thousands of candidate advertisements per search query. The accuracy of this estimation is important for revenue, but the cost of these computations represents a substantial expense, e.g., 10% to 30% of the total gross revenue. Caching the results of previous computations is a potential path to reducing this expense, but traditional domain-agnostic and revenue-agnostic approaches to do so result in substantial revenue loss. This paper presents three domain-specific caching mechanisms that successfully optimize for both factors. Simulations on a trace from the Bing advertising system show that a traditional cache can reduce cost by up to 27.7% but has negative revenue impact as bad as -14.1%. On the other hand, the proposed mechanisms can reduce cost by up to 20.6% while capping revenue impact between -1.3% and 0%. Based on Microsoft's earnings release for FY16 Q4, the traditional cache would reduce the net profit of Bing Ads by $84.9 to $166.1 million in the quarter, while our proposed cache could increase the net profit by $11.1 to $71.5 million.



Simplified workflow of how Bing advertising system serves ads to users.

Cachier: Edge-caching for recognition applications

Utsav Drolia, Katherine Guo, Jiaqi Tan, Rajeev Gandhi & Priya Narasimhan

The 37th IEEE International Conference on Distributed Computing Systems (ICDCS 2017), June 5 – 8, 2017, Atlanta, GA, USA

Recognition and perception-based mobile applications, such as image recognition, are on the rise. These applications recognize the user's surroundings and augment it with information and/or media. These applications are latency-sensitive. They have a soft-realtime nature - late results are potentially meaningless. On the one hand, given the compute-intensive nature of the tasks performed by such applications, execution is typically offloaded to the cloud. On the other hand, offloading such applications to the cloud incurs network latency, which can increase the user-perceived latency. Consequently, edge-computing has been proposed to let devices offload intensive tasks to edge-servers instead of the cloud, to reduce latency. In this paper, we propose a different

continued on page 12

continued from page 11

model for using edge-servers. We propose to use the edge as a specialized cache for recognition applications and formulate the expected latency for such a cache. We show that using an edge-server like a typical web-cache, for recognition applications, can lead to higher latencies. We propose Cachier, a system that uses the caching model along with novel optimizations to minimize latency by adaptively balancing load between the edge and the cloud by leveraging spatiotemporal locality of requests, using offline analysis of applications, and online estimates of network conditions. We evaluate Cachier for image-recognition applications and show that our techniques yield 3x speed-up in responsiveness, and perform accurately over a range of operating conditions. To the best of our knowledge, this is the first work that models edge-servers as caches for compute-intensive recognition applications, and Cachier is the first system that uses this model to minimize latency for these applications.

Carpool: A Bufferless On-Chip Network Supporting Adaptive Multicast and Hotspot Alleviation

Xiyue Xiang, Wentao Shi, Saugata Ghose, Lu Peng, Onur Mutlu & Nian-Feng Tzeng

In Proc. of the International Conference on Supercomputing (ICS), Chicago, IL, June 2017

Modern chip multiprocessors (CMPs) employ on-chip networks to enable communication between the individual cores. Operations such as coherence and synchronization generate a significant amount of the on-chip network traffic, and often create network requests that have one-to-many (i.e., a core multicasting a message to several cores) or many-to-one (i.e., several cores sending the same message to a common hotspot destination core) flows. As the number of cores in a CMP increases, one-to-many and many-to-one flows result in greater congestion on the network. To alleviate this congestion, prior work provides hardware support for efficient one-to-many and many-to-one flows in buffered on-chip networks. Unfortunately, this hardware support cannot be used in bufferless on-chip networks, which are shown to have lower hardware complexity and higher energy efficiency than buffered networks, and thus are likely a good fit for large-scale CMPs.

Andy Pavlo regales the crowd with tales of his database research at the 2017 PDL Spring Visit Day.

We propose Carpool, the first bufferless on-chip network optimized for one-to-many (i.e., multicast) and many-to-one (i.e., hotspot) traffic. Carpool is based on three key ideas: it (1) adaptively forks multicast flit replicas; (2) merges hotspot flits; and (3) employs a novel parallel port allocation mechanism within its routers, which reduces the router critical path latency by 5.7% over a bufferless network router without multicast support. We evaluate Carpool using synthetic traffic workloads that emulate the range of rates at which multithreaded applications inject multicast and hotspot requests due to coherence and synchronization. Our evaluation shows that for an 8×8 mesh network, Carpool reduces the average packet latency by 43.1% and power consumption by 8.3% over a bufferless network without multicast or hotspot support. We also find that Carpool reduces the average packet latency by 26.4% and power consumption by 50.5% over a buffered network with multicast support, while consuming 63.5% less area for each router.

Understanding Reduced-Voltage Operation in Modern DRAM Devices: Experimental Characterization, Analysis, and Mechanisms

Kevin K. Chang, A. Giray Yaglikçi, Saugata Ghose, Aditya Agrawal, Niladrish Chatterjee, Abhijith Kashyap, Donghyuk Lee, Mike O'Connor, Hasan Hassan & Onur Mutlu

Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS), Vol. 1, No. 1, June 2017.

The energy consumption of DRAM is a critical concern in modern computing systems. Improvements in manufacturing process technology have allowed DRAM vendors to lower the DRAM supply voltage conservatively, which reduces some of the DRAM energy consumption. We would like to reduce the DRAM supply voltage more aggressively, to further reduce energy. Aggressive supply voltage reduction requires a thorough understanding of the effect voltage scaling has on DRAM access latency and DRAM reliability.

In this paper, we take a comprehensive approach to understanding and exploiting the latency and reliability characteristics of modern DRAM when the supply voltage is lowered below the nominal voltage level specified by DRAM standards. Using an FPGA-based testing platform, we perform an experimental study of 124 real DDR3L (low-voltage) DRAM chips manufactured recently by three major DRAM vendors. We find that reducing the supply voltage below a certain point introduces bit errors in the data, and we comprehensively characterize the behavior of these errors. We discover that these errors can be avoided by increasing the latency of three major

continued on page 13

continued from page 12

DRAM operations (activation, restoration, and precharge). We perform detailed DRAM circuit simulations to validate and explain our experimental findings. We also characterize the various relationships between reduced supply voltage and error locations, stored data patterns, DRAM temperature, and data retention.

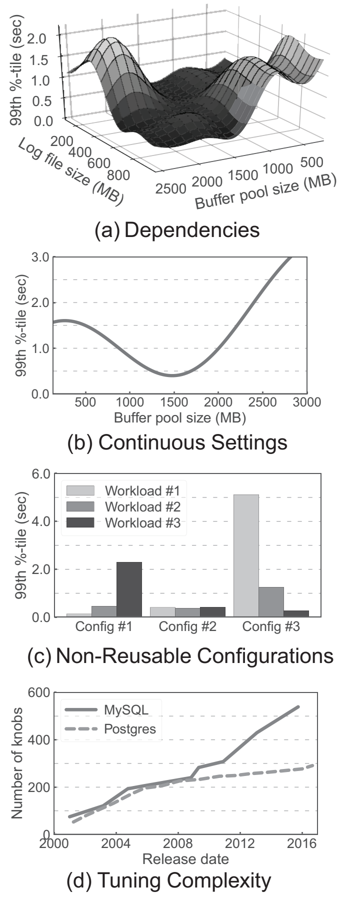
Based on our observations, we propose a new DRAM energy reduction mechanism, called Voltron. The key idea of Voltron is to use a performance model to determine by how much we can reduce the supply voltage without introducing errors and without exceeding a user-specified threshold for performance loss. Our evaluations show that Voltron reduces the average DRAM and system energy consumption by 10.5% and 7.3%, respectively, while limiting the average system performance loss to only 1.8%, for a variety of memory-intensive quad-core workloads. We also show that Voltron significantly outperforms prior dynamic voltage and frequency scaling mechanisms for DRAM.

Automatic Database Management System Tuning Through Large-scale Machine Learning

Dana Van Aken, Andrew Pavlo, Geoffrey J. Gordon & Bohan Zhang

ACM SIGMOD International Conference on Management of Data, May 14-19, 2017. Chicago, IL, USA.

Database management system (DBMS) configuration tuning is an essential aspect of any data-intensive application effort. But this is historically a difficult task because DBMSs have hundreds of configuration “knobs” that control everything in the system, such as the amount of memory to use for caches and how often data is written to storage. The problem with these knobs is that they are not standardized (i.e., two DBMSs use a different name for the same knob), not independent (i.e., changing one knob can impact others), and not universal (i.e., what works for one application may be suboptimal for another). Worse, information about the effects of the knobs typically comes only from (expensive) experience.



Motivating Examples – Figs. a to c show performance measurements for the YCSB workload running on MySQL (v5.6) using different configuration settings. Fig. d shows the number of tunable knobs provided in MySQL and Postgres releases over time.

To overcome these challenges, we present an automated approach that leverages past experience and collects new information to tune DBMS configurations: we use a combination of supervised and unsupervised machine learning methods to (1) select the most impactful knobs, (2) map unseen database workloads to previous workloads from which we can transfer experience, and (3) recommend knob settings. We implemented our techniques in a new tool called OtterTune and tested it on three DBMSs. Our evaluation shows that OtterTune recommends configurations that are as good as or better than ones generated by existing tools or a human expert.

Efficient Redundancy Techniques for Latency Reduction in Cloud Systems

Gauri Joshi, Emina Soljanin & Gregory Wornell

ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS) Volume 2 Issue 2, May 2017.

In cloud computing systems, assigning a task to multiple servers and waiting for the earliest copy to finish is an effective method to combat the variability in response time of individual servers and reduce latency. But adding redundancy may result in higher cost of computing resources, as well as an increase in queueing delay due to higher traffic load. This work helps in understanding when and how redundancy gives a cost-efficient reduction in latency. For a general task service time distribution, we compare different redundancy strategies in terms of the number of redundant tasks and the time when they are issued and canceled. We get the insight that the log-concavity of the task service time creates a dichotomy of when adding redundancy helps. If the service time distribution is log-convex (i.e., log of the tail probability is convex), then adding maximum redundancy reduces both latency and cost. And if it is log-concave (i.e., log of the tail probability is concave), then less redundancy, and early cancellation of redundant tasks is more effective. Using these insights, we design a general redundancy strat-

continued on page 14

continued from page 13

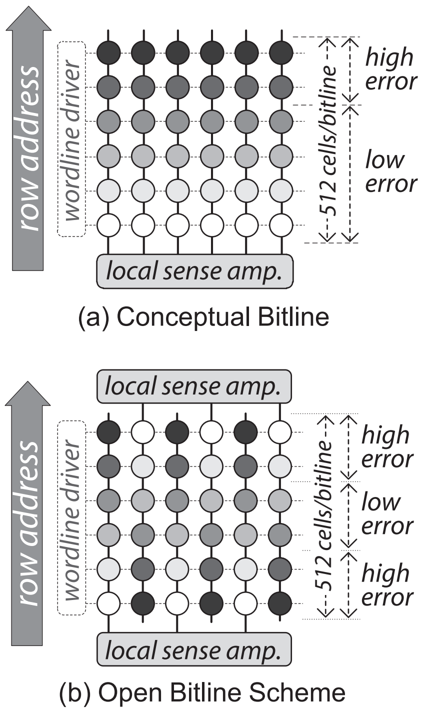
egy that achieves a good latency-cost trade-off for an arbitrary service time distribution. This work also generalizes and extends some results in the analysis of fork-join queues.

Design-Induced Latency Variation in Modern DRAM Chips: Characterization, Analysis, and Latency Reduction Mechanisms

Donghyuk Lee, Samira Khan, Lavanya Subramanian, Saugata Ghose, Rachata Ausavarungnirun, Gennady Pekhimenko, Vivek Seshadri & Onur Mutlu

Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS), Vol. 1, No. 1, June 2017.

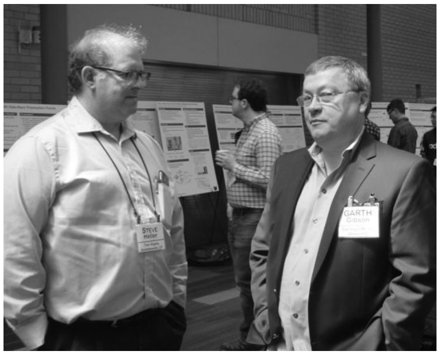
Variation has been shown to exist across the cells within a modern DRAM chip. Prior work has studied and exploited several forms of variation, such as manufacturing-process- or temperature-induced variation. We empirically demonstrate a new form of variation that exists within a real DRAM chip, induced by the design and placement of different components in the DRAM chip: different regions in DRAM, based on their relative distances from the peripheral structures, require different minimum access latencies for reliable operation. In particular, we show that in most real DRAM chips, cells closer to the peripheral structures can be accessed much faster than cells that are farther. We call this phenomenon design-induced variation in DRAM. Our goals are to i) understand design-induced variation that exists in real, state-of-the-art DRAM chips, ii) exploit it to develop low-cost mechanisms that can dynamically find and use the lowest latency at which to operate a DRAM chip reliably, and, thus, iii) improve overall system performance while ensuring reliable system operation.



(a) Conceptual Bitline

(b) Open Bitline Scheme

Design-Induced Variation Due to Row Organization



Garth Gibson chats with Steve Heller of Two Sigma during a poster session at the 2017 PDL Visit Day.

To this end, we first experimentally demonstrate and analyze designed-induced variation in modern DRAM devices by testing and characterizing 96 DIMMs (768 DRAM chips). Our characterization identifies DRAM regions that are vulnerable to errors, if operated at lower latency, and finds consistency in their locations across a given DRAM chip generation, due to design-induced variation. Based on our extensive experimental analysis, we develop two mechanisms that reliably reduce DRAM latency. First, DIVA Profiling uses runtime profiling to dynamically identify the lowest DRAM latency that does not introduce failures. DIVA Profiling exploits design-induced variation and periodically profiles only the vulnerable regions to determine the lowest DRAM latency at low cost. It is the first mechanism to dynamically determine the lowest latency that can be used to operate DRAM reliably. DIVA Profiling reduces the latency of read/write requests by 35.1%/57.8%, respectively, at 55°C. Our second mechanism, DIVA Shuffling, shuffles data such that values stored in vulnerable regions are mapped to multiple error-correcting code (ECC) codewords. As a result, DIVA Shuffling can correct 26% more multi-bit errors than conventional ECC. Combined together, our two mechanisms reduce read/write latency by 40.0%/60.5%, which translates to an overall system performance improvement of 14.7%/13.7%/13.8% (in 2-/4-/8-core systems) across a variety of workloads, while ensuring reliable operation.

Relaxed Operator Fusion for In-Memory Databases: Making Compilation, Vectorization, and Prefetching Work Together At Last

Prashanth Menon, Todd C. Mowry & Andrew Pavlo

Proceedings of the VLDB Endowment, Vol. 11, No. 1, 2017.

In-memory database management systems (DBMSs) are a key component of modern on-line analytic processing (OLAP) applications, since they provide low-latency access to large volumes of data. Because disk accesses are no longer the principle bottleneck in such systems, the focus in designing query execution engines has shifted to optimizing CPU performance. Recent systems have revived an older technique of using just-in-time (JIT)

continued on page 15

continued from page 14

compilation to execute queries as native code instead of interpreting a plan. The state-of-the-art in query compilation is to fuse operators together in a query plan to minimize materialization overhead by passing tuples efficiently between operators. Our empirical analysis shows, however, that more tactful materialization yields better performance.

We present a query processing model called "relaxed operator fusion" that allows the DBMS to introduce staging points in the query plan where intermediate results are temporarily materialized. This allows the DBMS to take advantage of inter-tuple parallelism inherent in the plan using a combination of prefetching and SIMD vectorization to support faster query execution on data sets that exceed the size of CPU-level caches. Our evaluation shows that our approach reduces the execution time of OLAP queries by up to 2.2X and achieves up to 1.8X better performance compared to other in-memory DBMSs.

EC-Cache: Load-Balanced, Low-Latency Cluster Caching with Online Erasure Coding

K. V. Rashmi, Mosharaf Chowdhury, Jack Kosaian, Ion Stoica & Kannan Ramchandran

12th USENIX Symposium on Operating Systems Design and Implementation, NOVEMBER 2–4, 2016, SAVANNAH, GA.

Data-intensive clusters and object stores are increasingly relying on in-memory object caching to meet the I/O performance demands. These systems routinely face the challenges of popularity skew, background load imbalance, and server failures, which result in severe load imbalance across servers and degraded I/O performance. Selective replication is a commonly used technique to tackle these challenges, where the number of cached replicas of an object is proportional to its popularity. In this paper, we explore an alternative approach using erasure coding.

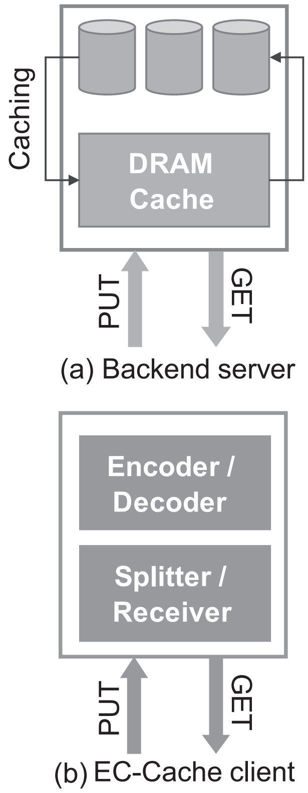
EC-Cache is a load-balanced, low latency cluster cache that uses online erasure coding to overcome the limitations of selective replication. EC-Cache employs erasure coding by: (i) splitting and erasure coding individual objects during writes, and (ii) late binding, wherein obtaining any k out of (k + r) splits of an object are sufficient, during reads. As compared to selective replication, EC-Cache improves load balancing by more than 3x and reduces the median and tail read latencies by more than 2x, while using the same amount of memory. EC-Cache does so using 10% additional bandwidth and a small increase in the amount of stored metadata. The benefits offered by EC-Cache are further amplified in the presence of background network load imbalance and server failures.

Having Your Cake and Eating It Too: Jointly Optimal Erasure Codes for I/O, Storage, and Network-bandwidth

KV Rashmi, Preetum Nakkiran, Jingyan Wang, Nihar B. Shah & Kannan Ramchandran

USENIX FAST, Feb 2015, Santa Clara, CA. Best paper.

Erasure codes, such as Reed-Solomon (RS) codes, are increasingly being deployed as an alternative to data-replication for fault tolerance in distributed storage systems. While RS codes provide significant savings in storage space, they can impose a huge burden on the I/O and network resources when reconstructing failed or otherwise unavailable data. A recent class of erasure codes, called minimum-storage-regeneration (MSR) codes, has emerged as a superior alternative to the popular RS codes, in that it minimizes network transfers during reconstruction while also being optimal with respect to storage and reliability. However, existing practical MSR codes do not address the increasingly important problem of I/O overhead incurred during reconstructions, and are, in general, inferior to RS codes in this regard. In this paper, we design erasure codes that are simultaneously optimal in terms of I/O, storage, and network bandwidth. Our design builds on top of a class of powerful practical codes, called the product-matrix-MSR codes. Evaluations show that our proposed design results in a significant reduction the number of I/Os consumed during reconstructions (a 5 reduction for typical parameters), while retaining optimality with respect to storage, reliability, and network bandwidth.



(a) Backend server

(b) EC-Cache client

Roles in EC-Cache: (a) backend servers manage interactions between caches and persistent storage for client libraries; (b) EC-Cache clients perform encoding and decoding during writes and reads.

continued from page 2

offs among them to maximize utility of the public and/or private cloud infrastructure.

» We are exploring new approaches to system support for large-scale machine learning. We have been especially active in investigating how ML systems should adapt to cloud computing environments. Important questions include how such systems should address dynamic resource availability, unpredictable levels of time-varying resource interference, and geo-distributed data. And, in a fun twist, we are developing new approaches to automatic tuning for ML systems—ML to improve ML.

Who We Are

Over the past 25 years, many faculty, staff, and students have been active members of the PDL. The lab has grown to over almost 90 current members (including staff, students and faculty), and research funding and output have seen similar growth while remaining focused on the PDL's and CMU's goals of delivering distinctive and top-quality education, fostering research, creativity and discovery, and using the new knowledge created on campus to serve our larger society.

PDL's first PhD graduate was Dr. Mark Holland (1994), who wrote his dissertation on 'On-Line Data Reconstruction in Redundant Disk Arrays.' Since then, dozens of PDL students have graduated with PhDs, Masters Degrees, and undergraduate degrees, and many have moved on to employment with PDL Consortium companies. We should also take a moment here to remember two PDL alumni who left us too soon: Howard Gobioff, PhD 1999, and Wittawat Tantisiriroj, PhD student.

Garth and Hugo Patterson at Hugo's graduation, 1997.

Our PDL members have garnered many prestigious awards, including 24 best paper awards. Numerous fellowships have been awarded to our graduate students, many from our sponsor companies. There have also been many faculty research awards. One of our first student members, Hugo Patterson (PhD '97), now a successful entrepreneur, has endowed a fellowship to support up and coming PDL entrepreneurs. The award provides academic support to a student working within the umbrella of the PDL who has had experience in the workforce prior to entering a graduate program.

Greg and some PDLers, from L to R, John Bucy, John Griffin, Andy Klosterman, Greg, and Jay Wylie.

In fact, Hugo is one of four PDL alums who have been named "Distinguished Alumni." Also honored are Howard Gobioff, PhD '99, Erik Riedel, PhD '99, and Bianca Schroeder, PhD '07.

The PDL Retreat

From the beginning, the PDL logo has included Skibo Castle, Andrew Carnegie's summer home near Dornoch, Scotland. In the past, it has represented "a fortress of storage" (like a redundant disk array). Later, it came to represent a "fortress of security" (à la self-securing devices). Perhaps, though, it is simply our vision of the ideal PDL Retreat venue.

The first official PDL Workshop and Retreat was held in October, 1993 for the purpose of interacting with our industry sponsors, offering them a chance to get to know the PDL researchers, hear about their work, offer feedback, and give the students a chance to develop relationships with prospective employers. At the first retreat, PDL research on disk arrays, parity logging and declustering was described by 20 CMU participants to 11 industry visitors from 6 sponsor companies.

One of the PDL students at the time recalls everyone wondering if they would have enough solid content to keep the industry attendees' attention throughout the retreat—of course it was not a problem. Every year since then, the difficult problem has been what to leave out, as the PDL researchers generate more cool ideas than will fit into the available time.

The first Retreat was held at the Hidden Valley Resort, PA. A number of retreats were held at the Nemacolin Woodlands Resort, in Farmington, PA. We now gather at the Omni Bedford Springs Resort, in Bedford, PA, and these days retreats are usually attended by over 100 participants.

In 1999 we also began holding an annual Spring Industry Visit Day. This is a one-day event that evolved as a result of requests from industry for more frequent interaction with PDL researchers.

Since its inception, Carnegie Mellon's Parallel Data Lab (PDL) has established itself as academia's premiere storage systems research center, consistently pushing the state-of-the-art with new storage system architectures, technologies, and design methodologies. In the words of our director "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." Let's see where the next 25 years take us!