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

PDL CONSORTIUM MEMBERS

Alibaba Group
Broadcom, Ltd.
Dell EMC
Facebook
Google
Hewlett Packard Enterprise
Hitachi, Ltd.
IBM Research
Intel Corporation
Micron
Microsoft Research
MongoDB
NetApp, Inc.
Oracle Corporation
Salesforce
Samsung Information Systems America
Seagate Technology
Toshiba
Two Sigma
Veritas
Western Digital

PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • SPRING 2018
http://www.pdl.cmu.edu/

Massive Indexed Directories in DeltaFS

by Qing Zheng, George Amvrosiadis & the DeltaFS Group

Faster storage media, faster interconnection networks, and improvements in systems software have significantly mitigated the effect of I/O bottlenecks in HPC applications. Even so, applications that read and write data in small chunks are limited by the ability of both the hardware and the software to handle such workloads efficiently. Often, scientific applications partition their output using one file per process. This is a problem on HPC computers with hundreds of thousands of cores and will only worsen with exascale computers, which will be an order of magnitude larger. To avoid wasting time creating output files on such machines, scientific applications are forced to use libraries that combine multiple I/O streams into a single file. For many applications where output is produced out-of-order, this must be followed by a costly, massive data sorting operation. DeltaFS allows applications to write to an arbitrarily large number of files, while also guaranteeing efficient data access without requiring sorting.

The first challenge when handling an arbitrarily large number of files is dealing with the resulting metadata load. We manage this using the transient and serverless DeltaFS file system [1]. The transient property of DeltaFS allows each program that uses it to individually control the amount of computing resources dedicated to the file system, effectively scaling metadata performance under application control. When combined with DeltaFS's serverless nature, file system design and provisioning decisions are decoupled from the overall design of the HPC platform. As a result, applications that create one file for each process are no longer tied to the platform storage system's ability to handle metadata-heavy workloads. The HPC platform can also provide scalable file creation rates without requiring a fundamental redesign of the platform's storage system.

The second challenge is guaranteeing both fast writing and reading for workloads that consist primarily of small I/O transfers. This work was inspired by interactions with cosmologists seeking to explore the trajectories of the highest energy particles in an astrophysics simulation using the VPIC plasma simulation code [2]. VPIC is a highly-optimized particle simulation code developed at Los Alamos National Laboratory (LANL). Each VPIC simulation proceeds in timesteps, and each process represents a bounding box in the physical simulation space that particles move through. Every few timesteps the simulation stops, and each process creates a file and writes the data for the particles that are currently located within its bounding box. This is the default, file-per-process

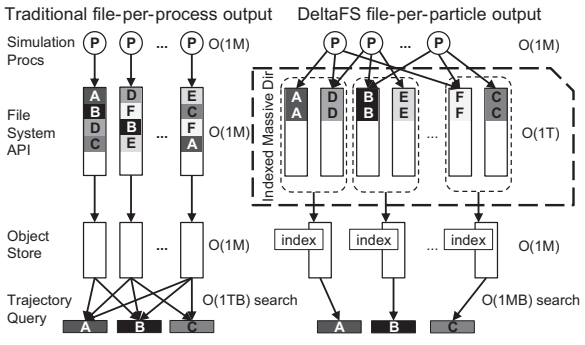


Figure 1: DeltaFS in-situ indexing of particle data in an Indexed Massive Directory. While indexed particle data are exposed as one DeltaFS subfile per particle, they are stored as indexed log objects in the underlying storage.

continued on page 11

FROM THE DIRECTOR'S CHAIR

GREG GANGER

Hello from fabulous Pittsburgh!

25 years! This past fall, we celebrated 25 years of the Parallel Data Lab. Started by Garth after he defended his PhD dissertation on RAID at UC-Berkeley, PDL has seen growth and success that I can't imagine he imagined… from the early days of exploring new disk array approaches to today's broad agenda of large-scale storage and data center infrastructure research… from a handful of core CMU researchers and industry participants to a vibrant community of scores of CMU researchers and 20 sponsor companies. Amazing.

It has been another great year for the Parallel Data Lab, and I'll highlight some of the research activities and successes below. Others, including graduations, publications, awards, etc., can be found throughout the newsletter. But, I can't not start with the biggest PDL news item of this 25th anniversary year: Garth has graduated;). More seriously, 25 years after founding PDL, including guiding/nurturing it into a large research center with sustained success (25 years!), Garth decided to move back to Canada and take the reins (as President and CEO) of the new Vector Institute for AI. We wish him huge success with this new endeavor! Garth has been an academic role model, a mentor, and a friend to me and many others… we will miss him greatly, and he knows that we will always have a place for him at PDL events.

Because it overlaps in area with Vector, I'll start my highlighting of PDL activities with our continuing work at the intersection for machine learning (ML) and systems. We continue to explore new approaches to system support for large-scale machine learning, especially aspects of how ML systems should adapt and be adapted in cloud computing environments. Beyond our earlier focus on challenges around dynamic resource availability and time-varying resource interference, we continue to explore challenges related to training models over geo-distributed data, training very large models, and how edge resources should be shared among inference applications using DNNs for video stream processing. We are also exploring how ML can be applied to make systems better, including even ML systems ;).

Indeed, much of PDL's expansive database systems research activities center on embedding automation in DBMSs. With an eye toward simplifying administration and improving performance robustness, there are a number of aspects of Andy's overall vision of a self-driving database system being explored and realized. To embody them, and other ideas, a new open source DBMS called Peloton has been created and is being continuously enhanced. There also continue to be cool results and papers on better exploitation of NVM in databases, improved concurrency control mechanisms, and range query filtering. I thoroughly enjoy watching (and participating) in the great energy that Andy has infused into database systems research at CMU.

Of course, PDL continues to have a big focus on storage systems research at various levels. At the high end, PDL's long-standing focus on metadata scaling for scalable storage has led to continued research into benefits of and approaches to allowing important applications to manage their own namespaces and metadata for periods of time. In addition to bypassing traditional metadata bottlenecks

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.

PARALLEL DATA LABORATORY

FACULTY

Greg Ganger (PDL Director)
412•268•1297
ganger@ece.cmu.edu

George Amvrosiadis
David Andersen
Lujo Bauer
Nathan Beckmann
Daniel Berger
Chuck Cranor
Lorrie Cranor
Christos Faloutsos
Kayvon Fatahalian
Rajeev Gandhi
Saugata Ghose
Phil Gibbons
Garth Gibson
Seth Copen Goldstein
Mor Harchol-Balter
Gauri Joshi
Todd Mowry
Onur Mutlu
Priya Narasimhan
David O'Hallaron
Andy Pavlo
Majd Sakr
M. Satyanarayanan
Srinivasan Seshan
Rashmi Vinayak
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

Jason Boles
Joan Digney
Chad Dougherty
Mitch Franzos
Alex Glikson
Charlene Zang

VISITING RESEARCHERS / POST DOCS

Rachata Ausavarungnirun
Hyeontaek Lim
Kazuhiro Saito

GRADUATE STUDENTS

Abutalib Aghayev
Joy Arulraj
Ben Blum
V. Parvathi Bhogaraju
Amirali Boroumand
Sol Boucher
Christopher Canel
Dominic Chen
Haoxian Chen
Malhar Chaudhari
Andrew Chung
Chris Fallin
Pratik Fegade
Ziqiang Feng
Samarth Gupta
Aaron Harlap
Kevin Hsieh
Fan Hu
Abhilasha Jain
Saksham Jain
Angela Jiang
Ellango Jothimurugesan
Saurabh Arun Kadekodi
Anuj Kalia
Rajat Kateja
Jin Kyu Kim
Thomas Kim
Vamshi Konagari
Jack Kosaian
Marcel Kost
Michael Kuchnik
Conglong Li
Kunmin Li
Yang Li
Yixin Luo
Lin Ma
Diptesh Majumdar
Ankur Mallick
Charles McGuffey
Prashanth Menon
Yuqing Miao
Wenqi Mou
Pooja Nilangekar
Yiqun Ouyang
Jun Woo Park
Aurick Qiao
Souptik Sen
Sivaprasad Sudhir
Aaron Tian
Dana Van Aken
Nandita Vijaykumar
Haoran Wang
Jianyu Wang
Justin Wang
Ziqi Wang
Jinliang Wei
Daniel Wong
Lin Xiao
Hao Zhang
Huanchen Zhang
Qing Zheng
Giulio Zhou

FROM THE DIRECTOR'S CHAIR

entirely during the heaviest periods of activity, this approach promises opportunities for efficient in-situ index creation to enable fast queries for subsequent analysis activities. At the lower end, we continue to explore how software systems should be changed to maximize the value from NVM storage, including addressing read-write performance asymmetry and providing storage management features (e.g., page-level checksums, dedup, etc.) without yielding load/store efficiency. We're excited about continuing to work with PDL companies on understanding where storage hardware is (and should be) going and how it should be exploited in systems.

PDL continues to explore questions of resource scheduling for cloud computing, which grows in complexity as the breadth of application and resource types grow. Our cluster scheduling research continues to explore how job runtime estimates can be automatically generated and exploited to achieve greater efficiency. Our most recent work explores more robust ways of exploiting imperfectly-estimated runtime information, finding that providing full distributions of likely runtimes (e.g., based on history of "similar" jobs) works quite well for real-world workloads as reflected in real cluster traces. We are also exploring scheduling for adaptively-sized "virtual clusters" within public clouds, which introduces new questions about which machine types to allocate, how to pack them, and how aggressively to release them.

I continue to be excited about the growth and evolution of the storage systems and cloud classes created and led by PDL faculty — their popularity is at an all-time high again this year. These project-intensive classes prepare 100s of MS students to be designers and developers for future infrastructure systems. They build FTLs that store real data (in a simulated NAND Flash SSD), hybrid cloud file systems that work, cluster schedulers, efficient ML model training apps, etc. It's really rewarding for us and for them. In addition to our lectures and the projects, these classes each feature 3-5 corporate guest lecturers (thank you, PDL Consortium members!) bringing insight on real-world solutions, trends, and futures.

Many other ongoing PDL projects are also producing cool results. For example, to help our (and others') file systems research, we have developed a new file system aging suite, called Geriatrix. Our key-value store research continues to expose new approaches to indexing and remote value access. 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.

The CMU fence displays a farewell message to Garth.

YEAR IN REVIEW

May 2018

- 20th annual Spring Visit Day.
- Qing Zheng and Michael Kuchnik will be interning with LANL this summer.

April 2018

- Andy Pavlo receive the 2018 Joel & Ruth Spira Teaching Award.
- Lorrie Cranor received the IAPP Leadership Award.
- Srinivasan Seshan was appointed Head of he Computer Science Dept. at CMU.
- Michael Kuchnik received an NDSEG Fellowship for his work on machine learning in HPC systems.
- Huanchen Zhang proposed his PhD research "Towards Space-Efficient High-Performance In-Memory Search Structures."
- Jun Woo Park presented "3Sigma: Distribution-based Cluster Scheduling for Runtime Uncertainty" at EuroSys '18 in Porto, Portugal.
- Charles McGuffey delivered his speaking skills talk on "Designing Algorithms to Tolerate Processor Faults."
- Qing Zheng gave his speaking skills talk "Light-Weight In-Situ Indexing For Scientific Workloads."

March 2018

- Andy Pavlo wins Google Faculty Research Award for his research on automatic database management systems.
- Anuj Kalia proposed his thesis research "Efficient Networked Systems for Datacenter Fabrics with RPCs."
- Nathan Beckmann presented "LHD: Improving Cache Hit Rate by Maximizing Hit Density" at NSDI '18 in Renton, WA.
- Rajat Kateja presented "Viyojit: Decoupling Battery and DRAM Capacities for Battery-Backed DRAM" at NVMW '18 in San Diego, CA.
- Rachata Ausavarungnirun presented "MASK: Redesigning the GPU Memory Hierarchy to Support Multi-Application Concurrency" at ASPLOS'18 in Williamsburg, VA.
- ASPLOS'18 in Williamsburg, VA.

Greg Ganger and PDL alums Hugo Patterson (Datrium) and Jiri Schindler (HPE) enjoy social time at the PDL Retreat.

February 2018

- Lorrie Cranor wins top SIGCHI Award, given to individuals who promote the application of human-computer interaction research to pressing social needs.
- Six posters were presented at the 1st SysML Conference at Stanford U. on various work related to creating more efficient systems for machine learning.
- Yixin Luo successfully defended his PhD dissertation on "Architectural Techniques for Improving NAND Flash Memory Reliability."
- Andy Pavlo awarded a Sloan Fellowship to continue his work on the study of database management systems, specifically main memory systems, non-relational systems (NoSQL), transaction processing systems (NewSQL) and large-scale data analytics.

December 2017

- Mor Harchol-Balter and Onur Mutlu were made Fellows of the ACM. Mor was selected "for contributions to performance modeling and analysis of distributed computing systems." Onur, who is now at ETH Zurich, was chosen for "contributions to computer architecture research, especially in memory systems."
- Joy Arulraj proposed his PhD research "The Design & Implementation of a Non-Volatile Memory Database Management System."
- Dana Van Aken gave her speaking skills talk on "Automatic Database Management System Tuning Through Large-scale Machine Learning."

November 2017

- Qing Zheng presented "Software-Defined Storage for Fast Trajectory Queries using a DeltaFS Indexed Massive Directory" at PDSW-DISCS '17 in Denver, CO.

October 2017

- Lorrie Cranor awarded FORE Systems Chair of Computer Science.
- Qing Zheng gave his speaking skills talk on "Light-weight In-situ Analysis with Frugal Resource Usage."
- Rachata Ausavarungnirun presented "Mosaic: A GPU Memory Manager with Application-Transparent Support for Multiple Page Sizes" and Vivek Seshadri presented Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology" at MICRO '17 in Cambridge, MA.
- Timothy Zhu presented "Workload Compactor: Reducing Datacenter Cost while Providing Tail Latency SLO Guarantees" at SoCC'17 in Santa Clara, CA.
- 25th annual PDL Retreat.

September 2017

- Garth Gibson to lead new Vector Institute for AI in Toronto.
- Hongyi Xin delivered his speaking skills talk on "Improving DNA Read Mapping with Error-resilient Seeds."

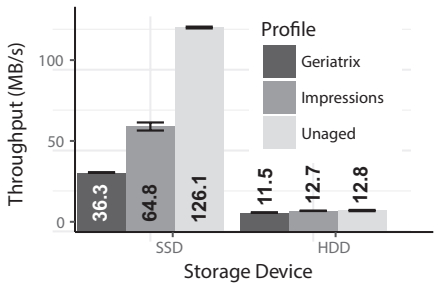
continued on page 32

Geriatrix: Aging what you see and what you don't see. A file system aging approach for modern storage systems

Saurabh Kadekodi, Vaishnavh Nagarajan, Gregory R. Ganger & Garth A. Gibson

2018 USENIX Annual Technical Conference (ATC'18). July 11–13, 2018, Boston, MA.

File system performance on modern primary storage devices (Flash-based SSDs) is greatly affected by aging of the free space, much more so than were mechanical disk drives. We introduce Geriatrix, a simple-to-use profile driven file system aging tool that induces target levels of fragmentation in both allocated files (what you see) and remaining free space (what you don't see), unlike previous approaches that focus on just the former. This paper describes and evaluates the effectiveness of Geriatrix, showing that it recreates both fragmentation effects better than previous approaches. Using Geriatrix, we show that measurements presented in many recent file systems papers are higher than should be expected, by up to 30% on mechanical (HDD) and up to 75% on Flash (SSD) disks. Worse, in some cases, the performance rank ordering of file system designs being compared are different from the published results.



Aging impact on Ext4 atop SSD and HDD. The three bars for each device represent the FS freshly formatted (unaged), aged with Geriatrix, and aged with Impressions. Although relatively small differences are seen with the HDD, aging has a big impact on FS performance on the SSD. Although their file fragmentation levels are similar, the higher free space fragmentation produced by Geriatrix induces larger throughput reductions than for Impressions.

Geriatrix will be released as open source software with eight built-in aging profiles, in the hopes that it can address the need created by the increased performance impact of file system aging in modern SSD-based storage.

A Case for Packing and Indexing in Cloud File Systems

Saurabh Kadekodi, Bin Fan, Adit Madan, Garth A. Gibson & Gregory R. Ganger

10th USENIX Workshop on Hot Topics in Cloud Computing. July 9, 2018, Boston, MA.

Tiny objects are the bane of highly scalable cloud object stores. Not only do tiny objects cause massive slowdowns, but they also incur tremendously high costs due to current operation-based pricing models. For example, in Amazon S3's current pricing scheme, uploading 1GB data by issuing tiny (4KB) PUT requests (at 0.0005 cents each) is approximately 57x more expensive than storing that same 1GB for a month. To address this problem, we propose client-side packing of files into gigabyte-sized blobs with embedded indices to identify each file's location. Experiments with a packing implementation in Alluxio (an open-source distributed file system) illustrate the potential benefits, such as simultaneously increasing file creation throughput by up to 61x and decreasing cost by over 99.99%.

SOAP: One Clean Analysis of All Age-Based Scheduling Policies

Ziv Scully, Mor Harchol-Balter & Alan Scheller-Wolf

Proceedings of ACM SIGMETRICS 2018 Conference on Measurement and Modeling of Computer Systems Los Angeles, CA, June 2018.

We consider an extremely broad class of M/G/1 scheduling policies called SOAP: Schedule Ordered by Age-based Priority. The SOAP policies include almost all scheduling policies in the literature as well as an infinite number of variants which have never been analyzed, or maybe not even conceived. SOAP policies range from classic policies, like first-come, first-serve (FCFS), foreground-background (FB), class-based priority, and shortest remaining processing time (SRPT); to much more complicated scheduling rules, such as the famously complex Gittins index policy and other policies in which a job's priority changes arbitrarily with its age. While the response time of policies in the former category is well understood, policies in the latter category have resisted response time analysis. We present a universal analysis of all SOAP policies, deriving the mean and Laplace-Stieltjes transform of response time.

Towards Optimality in Parallel Job Scheduling

Ben Berg, Jan-Pieter Dorsman & Mor Harchol-Balter

Proceedings of ACM SIGMETRICS 2018 Conference on Measurement and Modeling of Computer Systems Los Angeles, CA, June 2018.

To keep pace with Moore's law, chip designers have focused on increasing the number of cores per chip rather than single core performance. In turn, modern jobs are often designed to run on any number of cores. However, to effectively leverage these multi-core chips, one must address the question of how many cores to assign to each job. Given that jobs receive sublinear speedups from additional cores, there is an obvious tradeoff: allocating more cores to an individual job reduces the job's runtime, but in turn decreases

continued on page 6

continued from page 5

the efficiency of the overall system. We ask how the system should schedule jobs across cores so as to minimize the mean response time over a stream of incoming jobs.

To answer this question, we develop an analytical model of jobs running on a multi-core machine. We prove that EQUI, a policy which continuously divides cores evenly across jobs, is optimal when all jobs follow a single speedup curve and have exponentially distributed sizes. EQUI requires jobs to change their level of parallelization while they run. Since this is not possible for all workloads, we consider a class of "fixed-width" policies, which choose a single level of parallelization, k, to use for all jobs. We prove that, surprisingly, it is possible to achieve EQUI's performance without requiring jobs to change their levels of parallelization by using the optimal fixed level of parallelization, k*. We also show how to analytically derive the optimal k* as a function of the system load, the speedup curve, and the job size distribution.

In the case where jobs may follow different speedup curves, finding a good scheduling policy is even more challenging. In particular, we find that policies like EQUI which performed well in the case of a single speedup function now perform poorly. We propose a very simple policy, GREEDY*, which performs near-optimally when compared to the numerically-derived optimal policy.

3Sigma: Distribution-based Cluster Scheduling for Runtime Uncertainty

Jun Woo Park, Alexey Tumanov, Angela Jiang, Michael A. Kozuch & Gregory R. Ganger

EuroSys '18, April 23–26, 2018, Porto, Portugal. Supersedes CMU-PDL-17-107, Nov. 2017.

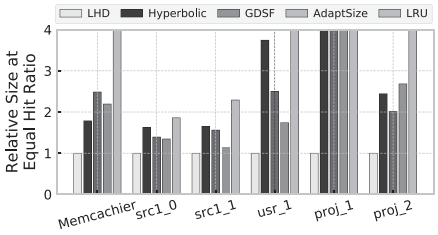
The 3Sigma cluster scheduling system uses job runtime histories in a new way. Knowing how long each job will execute enables a scheduler to more effectively pack jobs with diverse time concerns (e.g., deadline vs. the-sooner-the-better) and placement preferences on heterogeneous cluster resources. But, existing schedulers use single-point estimates (e.g., mean or median of a relevant subset of historical runtimes), and we show that they are fragile in the face of real-world estimate error profiles. In particular, analysis of job traces from three different large-scale cluster environments shows that, while the runtimes of many jobs can be predicted well, even state-of-the-art predictors have wide error profiles with 8–23% of predictions off by a factor of two or more. Instead of reducing relevant history to a single point, 3Sigma schedules jobs based on full distributions of relevant runtime histories and explicitly creates plans that mitigate the effects of anticipated runtime uncertainty. Experiments with workloads derived from the same traces show that 3Sigma greatly outperforms a state-of-the-art scheduler that uses point estimates from a state-of-the-art predictor; in fact, the performance of 3Sigma approaches the end-to-end performance of a scheduler based on a hypothetical, perfect runtime predictor. 3Sigma reduces SLO miss rate, increases cluster goodput, and improves or matches latency for best effort jobs.

LHD: Improving Cache Hit Rate by Maximizing Hit Density

Nathan Beckmann, Haoxian Chen & Asaf Cidon

15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18). April 9–11, 2018, Renton, WA.

Cloud application performance is heavily reliant on the hit rate of datacenter key-value caches. Key-value caches typically use least recently used (LRU) as their eviction policy, but LRU's hit rate is far from optimal under real workloads. Prior research has proposed many eviction policies that improve on LRU, but these policies make restrictive assumptions that hurt their hit rate, and they can be difficult to implement efficiently.



Relative cache size needed to match LHD's hit rate on different traces. LHD requires roughly one-fourth of LRU's capacity, and roughly half of that of prior eviction policies.

We introduce least hit density (LHD), a novel eviction policy for key-value caches. LHD predicts each object's expected hits-per-space-consumed (hit density), filtering objects that contribute little to the cache's hit rate. Unlike prior eviction policies, LHD does not rely on heuristics, but rather rigorously models objects' behavior using conditional probability to adapt its behavior in real time.

To make LHD practical, we design and implement RankCache, an efficient key-value cache based on memcached. We evaluate RankCache and LHD on commercial memcached and enterprise storage traces, where LHD consistently achieves better hit rates than prior policies. LHD requires much less space than prior policies to match their hit rate, on average 8X less than LRU and 2–3X less than recently proposed policies. Moreover, RankCache requires no synchronization in the common case, improving request throughput at 16 threads by 8 over LRU and by 2X over CLOCK.

continued on page 7

continued from page 6

Tributary: Spot-dancing for Elastic Services with Latency SLOs

Aaron Harlap, Andrew Chung, Alexey Tumanov, Gregory R. Ganger & Phillip B. Gibbons

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-18-102, Jan. 2018.

The Tributary elastic control system embraces the uncertain nature of transient cloud resources, such as AWS spot instances, to manage elastic services with latency SLOs more robustly and more cost-effectively. Such resources are available at lower cost, but with the proviso that they can be preempted en masse, making them risky to rely upon for business-critical services. Tributary creates models of preemption likelihood and exploits the partial independence among different resource offerings, selecting collections of resource allocations that will satisfy SLO requirements and adjusting them over time as client workloads change. Although Tributary's collections are often larger than required in the absence of preemptions, they are cheaper because of both lower spot costs and partial refunds for preempted resources. At the same time, the often-larger sets allow unexpected workload bursts to be handled without SLO violation. Over a range of web service workloads, we find that Tributary reduces cost for achieving a given SLO by 81–86% compared to traditional scaling on non-preemptible resources and by 47–62% compared to the high-risk approach of the same scaling with spot resources.

MLtuner: System Support for Automatic Machine Learning Tuning

Henggang Cui, Gregory R. Ganger & Phillip B. Gibbons

arXiv:1803.07445v1 [cs.LG] 20 Mar, 2018.

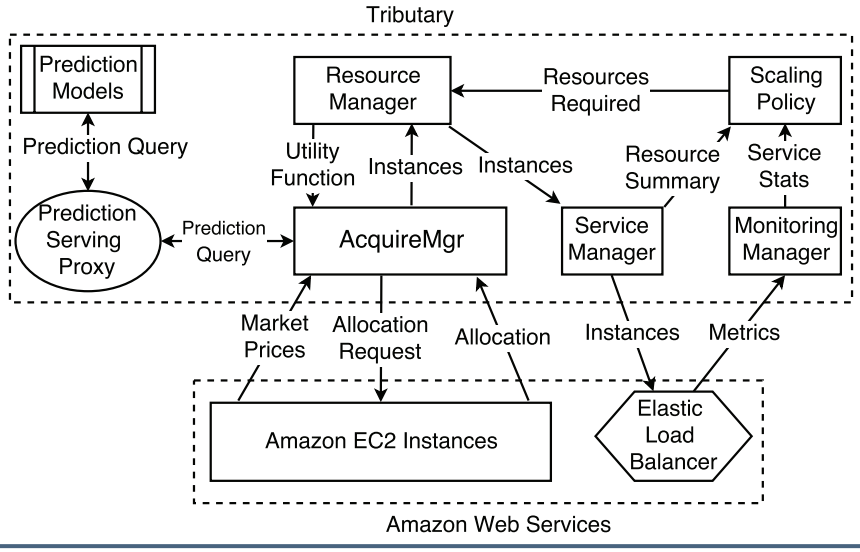
MLtuner automatically tunes settings for training tunables — such as the learning rate, the momentum, the mini-batch size, and the data staleness bound—that have a significant impact on large-scale machine learning (ML) performance. Traditionally, these tunables are set manually, which is unsurprisingly error prone and difficult to do without extensive domain knowledge. MLtuner uses efficient snapshotting, branching, and optimization-guided online trial-and-error to find good initial settings as well as to re-tune settings during execution. Experiments show that MLtuner can robustly find and re-tune tunable settings for a variety of ML applications, including image classification (for 3 models and 2 datasets), video classification, and matrix factorization. Compared to state-of-the-art ML auto-tuning approaches, MLtuner is more robust for large problems and over an order of magnitude faster.

Addressing the Long-Lineage Bottleneck in Apache Spark

Haoran Wang, Jinliang Wei & Garth Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-18-101, January 2018.

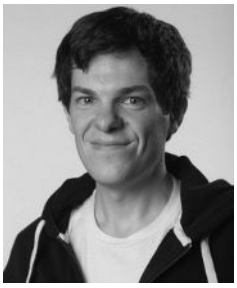
Apache Spark employs lazy evaluation [11, 6]; that is, in Spark, a dataset is represented as Resilient Distributed Dataset (RDD), and a single-threaded application (driver) program simply describes transformations (RDD to RDD), referred to as lineage [7, 12], without performing distributed computation until output is requested. The lineage traces computation and dependency back to external (and assumed durable) data sources, allowing Spark to opportunistically cache intermediate RDDs, because it can recompute everything from external data sources. To initiate computation on worker machines, the driver process constructs a directed acyclic graph (DAG) representing computation and dependency according to the requested RDD's lineage. Then the driver broadcasts this DAG to all involved workers requesting they execute their portion of the result RDD. When a requested RDD has a long lineage, as one would expect from iterative convergent or streaming applications [9, 15], constructing and broadcasting computational dependencies can become a significant bottleneck. For example, when solving matrix factorization using Gemulla's iterative convergent



The Tributary Architecture.

continued on page 20

April 2018
Andy Pavlo Receives 2018 Joel & Ruth Spira Teaching Award



The School of Computer Science honored outstanding faculty and staff members April 5 during the annual Founder's Day ceremony in Rashid Auditorium. It was the seventh year for the event and was hosted by Dean Andrew Moore. Andy Pavlo, Assistant Professor in the Computer Science Department (CSD), was the winner of the Joel and Ruth Spira Teaching Award, sponsored by Lutron Electronics Co. of Coopersburg, Pa., in honor of the company's founders and the inventor of the electronic dimmer switch.

-- CMU SCS news, April 5, 2018

April 2018
Lorrie Cranor Receives IAPP Leadership Award

Lorrie Cranor has received the 2018 Leadership Award from The International Association of Privacy Professionals (IAPP). Cranor, a professor in the Institute for Software Research and the Department of Engineering and Public Policy, accepted the award at the IAPP's Global Privacy Summit on March 27. "Lorrie Cranor, for 20 years, has been a leading voice and a leader in the privacy field," said IAPP President and CEO Trevor Hughes. "She developed some of the earliest privacy enhancing technologies, she developed a groundbreaking program at Carnegie Mellon University to create future generations of privacy engineers, and she has been a steadfast supporter, participant and leader of the field of privacy for that entire time. Her merits as recipient for our privacy leadership award are unimpeachable. She's as great a person as we have in our world." The IAPP Leadership Award is given annually to individuals who demonstrate an "ongoing commitment to furthering privacy policy, promoting recognition of privacy issues and advancing the growth and visibility of the privacy profession." Cranor helped develop and is now co-director of CMU's MSIT-Privacy Engineering master's degree program as well as director of the CyLab Usable Privacy and Security Laboratory.

--CMU Piper, April 5, 2018

April 2018
Welcome Baby Nora!

Pete and Laura Losi, and Grandma Karen Lindenfelser are thrilled to announce Nora Grace joined big sister Layla Anne and big cousin Landon Thomas to become a family of four (five if you count, Rudy, the granddog). Nora was born Friday the 13th at 11:50 am at 7 lbs 19.5 inches.

April 2018
Srinivasan Seshan Appointed Head of CSD

Srinivasan Seshan has been appointed head of the Computer Science Department (CSD), effective July 1. He succeeds Frank Pfenning, who will re-

turn to full-time teaching and research. "We are all excited about Srini Seshan's new role as head of CSD," said School of Computer Science Dean Andrew Moore. "He is an outstanding researcher and teacher, and I'm confident that his expanded role in leadership will help the department reach even greater heights." Seshan joined the CSD faculty in 2000, and served as the department's associate head for graduate education from 2011 to 2015. His research focuses on improving the design, performance and security of computer networks, including wireless and mobile networks. He earned his bachelor's, master's and doctoral degrees in computer science at the University of California, Berkeley. He worked as a research staff member at IBM's T.J. Watson Research Center for five years before joining Carnegie Mellon.

--CMU Piper, April 5, 2018

March 2018
Andy Pavlo Wins Google Faculty Research Award

The CMU Database Group and the PDL are pleased to announce that Prof. Andy Pavlo has won a 2018 Google Faculty Research Award. This award was for his research on automatic database management systems. Andy was one of a total 14 faculty members at Carnegie Mellon University selected for this award.

The Google Faculty Research Awards is an annual open call for proposals on computer science and related topics such as machine learning, machine perception, natural language processing, and quantum computing. Grants cover tuition for a graduate student and provide both faculty and students the opportunity to work directly with Google researchers and engineers.

continued on page 9

continued from page 8

This round received 1033 proposals covering 46 countries and over 360 universities from which 152 were chosen to fund. The subject areas that received the most support this year were human computer interaction, machine learning, machine perception, and systems.

-- Google and CMU Database Group News, March 20, 2018

February 2018
Lorrie Cranor Wins Top SIGCHI Award

Lorrie Cranor, a professor in the Institute for Software Research and the Department of Engineering and Public Policy, is this year's recipient of the Social Impact Award from the Association for Computing Machinery Special Interest Group on Computer Human Interaction (SIGCHI).

The Social Impact Award is given to mid-level or senior individuals who promote the application of human-computer interaction research to pressing social needs and includes an honorarium of $5,000, the opportunity to give a talk about the awarded work at the CHI conference, and lifetime invitations to the annual SIGCHI award banquet.

"Lorrie's work has had a huge impact on the ability of non-technical users to protect their security and privacy through her user-centered approach to security and privacy research and development of numerous tools and technologies," said Blase Ur, who prepared Lorrie's nomination. Ur is a former Ph.D. student of Lorrie's, and is now an assistant professor at the University of Chicago.

In addition to Ur, three former students from Cranor's CyLab Usable Privacy and Security Lab – Michelle Mazurek, Florian Schaub and Yang Wang – supported Lorrie's nomination. "All four of us are currently assistant professors, spread out across the United States," said Ur, who received his doctorate degree in 2016. "In addition to this impact on end users, the four of us who jointly nominated her have also benefitted greatly from her mentorship."

A full summary of this year's SIGCHI award recipients can be found on the organization's website.

-- info from Cylab News, Daniel Tkacik, Feb. 23, 2018

February 2018
Andy Pavlo Awarded a Sloan Fellowship

"The Sloan Research Fellows represent the very best science has to offer," said Sloan President Adam Falk. "The brightest minds, tackling the hardest problems, and succeeding brilliantly — fellows are quite literally the future of 21st century science."

Andrew Pavlo, an assistant professor of computer science, specializes in the study of database management systems, specifically main memory systems, non-relational systems (NoSQL), transaction processing systems (NewSQL) and large-scale data analytics. He is a member of the Database Group and the Parallel Data Laboratory. He joined the Computer Science Department in 2013 after earning a Ph.D. in computer science at Brown University. He won the 2014 Jim Gray Doctoral Dissertation Award from the Association for Computing Machinery's (ACM) Special Interest Group on the Management of Data.

-- Carnegie Mellon University News, Feb. 15, 2018

December 2017
Welcome Baby Sebastian!

In not-unexpected news, David, Erica and big sister Aria are delighted to announce the arrival of a squirmy and very snuggly addition to their family. Sebastian Alexander Andersen-Fuchs was born December 11, 2017, at 11:47 am at 8lb 8oz and 21" long. Mom and baby are healthy, and Aria is very excited to be a big sister.

December 2017
Mor Harchol-Balter and Onur Mutlu Fellows of the ACM

Congratulations to Mor (Professor of CS) and Onur (adjunct Professor of ECE), who have been made Fellows of the ACM.

From the ACM website: "To be selected as a Fellow is to join our most renowned member grade and an elite group that represents less than 1 percent of ACM's overall membership," explains ACM President Vicki L. Hanson. "The Fellows program allows us to shine a light on landmark contributions to computing, as well as the men and women whose hard work, dedication, and inspiration are responsible for groundbreaking work that improves our lives in so many ways."

Mor was selected "for contributions to performance modeling and analysis of distributed computing systems."

Onur, who is now at ETH Zurich, was chosen for "contributions to computer architecture research, especially in memory systems."

--with info from www.acm.org

continued on page 10

AWARDS & OTHER PDL NEWS

continued from page 9

November 2017
Welcome Baby Will!

Kevin Hsieh and his wife would like share the news of their new baby! Will was born on November 15, 2017 at 11:15am (not a typo…). He was born at 6lb 7oz and 20'' long. Since then, he has been growing very well and keeping his family busy.

October 2017
Welcome Baby Jonas!

Jason & Chien-Chiao Boles are excited to announce the arrival of their son Jonas at 7:42pm, October 18th. Jonas was born a few weeks early — a surprise for us all. Everyone is doing well so far.

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, became 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
Satya Honored for Creation of Andrew File System

The Association for Computing Machinery has named the developers of CMU'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 CS 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.

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

continued from page 1

mode of VPIC. For each timestep, 40 bytes of data is produced per particle representing the particle's spatial location, velocity, energy, etc. We refer to the entire particle data written at the same timestep as a frame, because frame data is often used by domain scientists to construct false-color movies of the simulation state over time. Large-scale VPIC simulations have been conducted with up to trillions of particles, generating terabytes of data for each frame.

Domain scientists are often interested in a tiny subset of particles with specific characteristics, such as high energy, that is not known until the simulation ends. All data for each such particle is gathered for further analysis, such as visualizing its trajectory through space over time. Unfortunately, particle data within a frame is written out of order, since output order depends on the particles' spatial location. Therefore, in order to locate individual particles' data over time, all output data must be sorted before they can be analyzed.

For scientists working with VPIC, it would be significantly easier programmatically to create a separate file for each particle, and append a 40-byte data record on each timestep. This would reduce analysis queries to sequentially reading the contents of a tiny number of particle files. Attempting to do this in today's parallel file systems, however, would be disastrous for performance. Expecting existing HPC storage stacks and file systems to adapt to scientific needs such as this one, however, is lunacy. Parallel file systems are designed to be long-running, robust services that work across applications. They are typically kernel resident, mainly developed to manage the hardware, and primarily optimized for large sequential data access. DeltaFS aims to provide this file-per-particle representation to applications, while ensuring that storage hardware is utilized to its full performance potential. A comparison of the file-per-process (current state-of-the-art) and file-per-particle (DeltaFS) representations is shown in Figure 1.

To improve the performance of applications with small I/O access patterns similar to VPIC, we propose an Indexed Massive Directory — a new technique for indexing data in-situ as it is written to storage. In-situ indexing of massive amounts of data written to a single directory simultaneously, and in an arbitrarily large number of files with the goal of efficiently recalling data written to the same file without requiring any time-consuming data post-processing steps to reorganize it. This greatly improves the readback performance of applications, at the price of small overheads associated with partitioning and indexing the data during writing. We achieve this through a memory-efficient indexing mechanism for reordering and indexing data, and a log-structured storage layout to pack small writes into large log objects, all while ensuring compute node resources are used frugally.

We evaluated the efficiency of the Indexed Massive Directory on LANL's Trinity hardware (Figure 2). By applying in-situ partial sorting of VPIC's particle output, we demonstrated over 5000x speedup in reading a single particle's trajectory from a 48- billion particle simulation output using only a single CPU core, compared to post-processing the entire dataset (10TiB) using the same amount of CPU cores as the original simulation. This speedup increases with simulation scale, while the total memory used for partial sort is fixed at 3% of the memory available to the simulation code. The cost of this read acceleration is the increased work in the in-situ pipeline and the additional storage capacity dedicated to storing the indexes. These results are encouraging, as they indicate that the output write buffering stage of the software-defined storage stack can be leveraged for one or more forms of efficient in-situ analysis, and can be applied to more kinds of query workloads.

For more information, please see [3] or visit our project page at www.pdl.cmu.edu/DeltaFS/

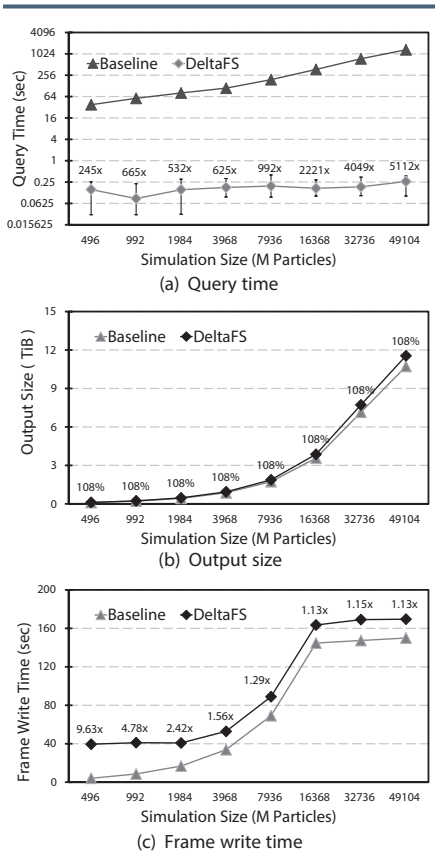


Figure 2: Results from real VPIC simulation runs with and without DeltaFS at LANL Trinity computer.

References

[1] Zheng, Q., Ren, K., Gibson, G., Settlemyer, B. W., and Grider, G. DeltaFS: Exascale file systems scale better without dedicated servers. In Proceedings of the 10th Parallel Data Storage Workshop (PDSW 15), pp. 1–6.

[2] Byna, S., Sisneros, R., Chadalavada, K., and Koziol, Q. Tuning parallel I/O on blue waters for writing 10 trillion particles. In Cray User Group (CUG) (2015).

[3] Qing Zheng, George Amvrosiadis, Saurabh Kadekodi, Garth Gibson, Chuck Cranor, Brad Settlemyer, Gary Grider, Fan Guo. Software-Defined Storage for Fast Trajectory Queries using a DeltaFS Indexed Massive Directory. PDSW-DISCS 2017, Denver, CO, November 2017.

Jun Woo Park, Greg Ganger and PDL 3Sigma Group

3Sigma: Distribution-Based Cluster Scheduling for Runtime Uncertainty

Modern cluster schedulers face a daunting task. Modern clusters support a diverse mix of activities, including exploratory analytics, software development and test, scheduled content generation, and customer-facing services [2]. Pending work is typically mapped to heterogeneous resources to satisfy deadlines for business-critical jobs, minimize delays for interactive best-effort jobs, maximize efficiency, and so on. Cluster schedulers are expected to make that happen.

Knowledge of the runtimes of these pending jobs has been identified as a powerful building block for modern cluster schedulers. With it, a scheduler can pack jobs more aggressively in a cluster's resource assignment plan, for instance by allowing a latency-sensitive best-effort job to run before a high-priority batch job provided that the priority job will still meet its deadline. Runtime knowledge allows a scheduler to determine whether it is better to start a job immediately on suboptimal machine types with worse expected performance, wait for the jobs currently occupying the preferred machines to finish, or to preempt them. Exploiting job runtime knowledge leads to better, more robust scheduler decisions than relying on hard-coded assumptions.

In most cases, the job runtime estimates are based on previous runtimes observed for similar jobs (e.g., from the same user or by the same periodic job script). When such estimates are accurate, the schedulers relying on them outperform those using other approaches.

However, we find that estimate errors, while expected in large, multi-use clusters, cover an unexpectedly larger range. Applying a state-of-the-art ML-based predictor [1] to three real-world traces, including the well-studied Google cluster trace [2] and new traces from data analysis clusters used at a hedge fund and a scientific site, shows good estimates in general (e.g., 77–92% within a factor of two of the actual runtime and most much closer). Unfortunately, 8–23% are not within that range, and some are off by an order of magnitude or more. Thus, a significant percentage of runtime estimates will be well outside the error ranges previously reported. Worse, we find that schedulers relying on runtime estimates cope poorly with such error profiles. Comparing the middle two bars of Fig. 1 shows one example of how much worse a state-of-the-art scheduler does with real estimate error profiles as compared to having perfect estimates.

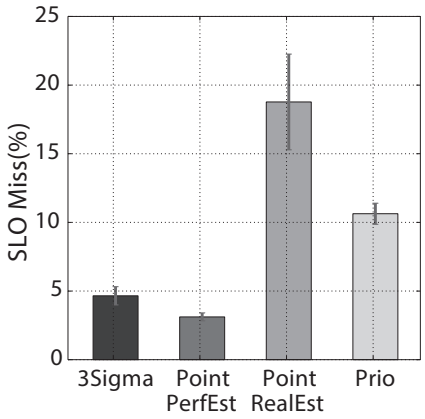


Figure 1: Comparison of 3Sigma with three other scheduling approaches w.r.t. SLO (deadline) miss rate, for a mix of SLO and best effort jobs derived from the Google cluster trace [2] on a 256-node cluster. 3Sigma, despite estimating runtime distributions online with imperfect knowledge of job classification, approaches the performance of a hypothetical scheduler using perfect runtime estimates (PointPerfEst). Full historical runtime distributions and mis-estimation handling helps 3Sigma outperform PointRealEst, a state-of-the-art point-estimate-based scheduler. The value of exploiting runtime information, when done well, is confirmed by comparison to a conventional priority-based approach (Prio).

Our 3Sigma cluster scheduling system uses all of the relevant runtime history for each job rather than just a point estimate derived from it. Instead, it uses expected runtime distributions (e.g., the histogram of observed runtimes), taking advantage of the much richer information (e.g., variance, possible multi-modal behaviors, etc.) to make more robust decisions. The first bar of Fig. 1 illustrates 3Sigma's efficacy.

By considering the range of possible runtimes for a job, and their likelihoods, 3Sigma can explicitly consider the various potential outcomes from each possible plan and select a plan based on optimizing the expected outcome. For example, the predicted distribution for one job might have low variance, indicating that the scheduler can be aggressive in packing it in, whereas another job's high variance might suggest that it should be scheduled early (relative to its deadline). 3Sigma similarly exploits the runtime distribution to adaptively address the problem of point over-estimates, which may suggest that the scheduler will avoid scheduling a job based on the likelihood of missing its deadline.

In application, 3Sigma replaces the scheduling component of a cluster manager (e.g. YARN). The cluster manager remains responsible for job and resource life-cycle management.

Job requests are received asynchronously by 3Sigma from the cluster manager (Step 1 of Fig. 2). As is typical for such systems, the specification of the request includes a number of attributes, such as (1) the name of the job to be run, (2) the type of job to be run (e.g. MapReduce), (3) the user submitting the job, and (4) a specification of the resources requested.

continued on page 13

continued from page 12

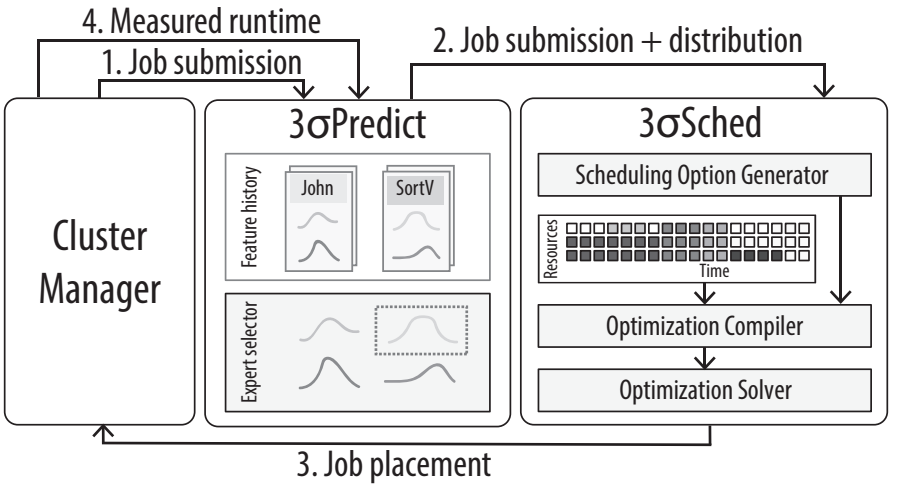


Figure 2: End-to-end system integration

The role of the predictor component 3σPredict is to provide the core scheduler with a probability distribution of the execution time of the submitted job. 3σPredict does this by maintaining a history of previously executed jobs, identifying a set of jobs that, based on their attributes, are similar to the current job and deriving the runtime distribution the selected jobs' historical runtimes (Step 2 of Fig. 2). Given a distribution of expected job runtimes and request specifications, the core scheduler, 3σSched decides which jobs to place on which resources and when. The scheduler evaluates the expected utility of each option and the expected resource consumption and availability over the scheduling horizon. Valuations and computed resource capacity are then compiled into an optimization problem, which is solved by an external solver. 3σSched translates the solution into an updated schedule and submits the schedule to the cluster manager (Step 3 of Fig. 2). On completion, the job's actual runtime is recorded by 3σPredict (along with the attribute information from the job) and incorporated into the job history for future predictions (Step 4 of Fig. 2).

Full system and simulation experiments with production-derived workloads demonstrate 3Sigma's effectiveness. Using its imperfect but automatically-generated history-based runtime distributions, 3Sigma outperforms both a state-of-the-art point-estimate-based scheduler and a priority-based (runtime-unaware) scheduler, especially for mixes of deadline-oriented jobs and latency-sensitive jobs on heterogeneous resources. 3Sigma simultaneously provides higher (1) SLO attainment for deadline-oriented jobs and (2) cluster goodput (utilization).

Our evaluation of 3Sigma, yielded five key takeaways. First, 3Sigma achieves significant improvement over the state-of-the-art in SLO miss rate, best-effort job goodput, and best-effort latency in a fully-integrated real cluster deployment, approaching the performance of the unrealistic PointPerfEst in SLO miss rate and BE latency. Second, all of the 3σSched component features are important, as seen via a piecewise benefit attribution. Third, estimated distributions are beneficial in scheduling even if they are somewhat inaccurate, and such inaccuracies are better handled by distribution-based scheduling than point-estimate-based scheduling. In fact, experiments with trace-derived workloads both on a real 256-node cluster and in simulation demonstrate that 3Sigma's distribution-based scheduling greatly outperforms a state-of-the-art point-estimate scheduler, approaching the performance of a hypothetical scheduler operating with perfect runtime estimates. Fourth, 3Sigma performs well (i.e., comparably to PointPerfEst) under a variety of conditions, such as varying cluster load, relative SLO job deadlines, and prediction inaccuracy. Fifth, we show that the 3Sigma components (3σPredict and 3σSched) can scale to >10000 nodes. Overall, we see that 3Sigma robustly exploits runtime distributions to improve SLO attainment and best-effort performance, dealing gracefully with the complex runtime variations seen in real cluster environments.

For more information, please see [3] or visit www.pdl.cmu.edu/TetriSched/

References

[1] Alexey Tumanov, Angela Jiang, Jun Woo Park, Michael A. Kozuch, and Gregory R. Ganger. 2016. JamaisVu: Robust Scheduling with AutoEstimated Job Runtimes. Technical Report CMU-PDL-16-104. Carnegie Mellon University.

[2] Charles Reiss, Alexey Tumanov, Gregory R. Ganger, Randy H. Katz, and Michael A. Kozuch. 2012. Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis. In Proc. of the 3nd ACM Symposium on Cloud Computing (SOCC '12).

[3] Jun Woo Park, Alexey Tumanov, Angela Jiang, Michael A. Kozuch, Gregory R. Ganger. 3Sigma: Distribution-based Cluster Scheduling for Runtime Uncertainty. EuroSys '18, April 23–26, 2018, Porto, Portugal.

DISSERTATION ABSTRACT: Architectural Techniques for Improving NAND Flash Memory Reliability

Yixin Luo
Carnegie Mellon University, SCS
PhD Defense — February 9, 2018

Raw bit errors are common in NAND flash memory and will increase in the future. These errors reduce flash reliability and limit the lifetime of a flash memory device. This dissertation improves flash reliability with a multitude of low-cost architectural techniques. We show that NAND flash memory reliability can be improved at low cost and with low performance overhead by deploying various architectural techniques that are aware of higher-level application behavior and underlying flash device characteristics.

This dissertation analyzes flash error characteristics and workload behavior through rigorous experimental characterization and designs new flash controller algorithms that use the insights gained from our analysis to improve flash reliability at low cost. We investigate four novel directions. (1) We propose a new technique called WARM that improves flash lifetime by 12.9 times by managing flash retention differently for write-hot data and write-cold data. (2) We propose a new framework that learns an online flash channel model for each chip and enables four new flash controller algorithms to improve flash write endurance by up to 69.9%. (3) We identify three new error characteristics in 3D NAND flash memory through comprehensive experimental characterization of real 3D NAND chips, and propose four new techniques that mitigate these new errors and improve 3D NAND raw bit error rate by up to 66.9%. (4) We propose a new technique called HeatWatch that improves 3D NAND lifetime by 3.85 times by utilizing the self-healing effect to mitigate retention errors in 3D NAND.

Industry guests and CMU folks boarding the bus to head to Bedford Springs for the PDL Retreat

DISSERTATION ABSTRACT: Fast Storage for File System Metadata

Kai Ren
Carnegie Mellon University, SCS
PhD Defense — August 8, 2017

In an era of big data, the rapid growth of data that many companies and organizations produce and manage continues to drive efforts to improve the scalability of storage systems. The number of objects presented in storage systems continue to grow, making metadata management critical to the overall performance of file systems. Many modern parallel applications are shifting toward shorter durations and larger degree of parallelism. Such trends continue to make storage systems to experience more diverse metadata intensive workloads.

The goal of this dissertation is to improve metadata management in both local and distributed file systems. The dissertation focuses on two aspects. One is to improve the out-of-core representation of file system metadata, by exploring the use of log-structured multi-level approaches to provide a unified and efficient representation for different types of secondary storage devices (e.g., traditional hard disk and solid state disk). We have designed and implemented TableFS and its improved version SlimFS, which shows 50% to 10x faster than traditional Linux file systems. The other aspect is to demonstrate that such representation also can be flexibly integrated with many namespace distribution mechanisms to scale metadata performance of distribution file systems, and provide better support for a variety of big data applications in data center environment. Our distributed metadata middleware IndexFS can help improve metadata performance for PVFS, Lustre and HDFS by scaling to as many as 128 metadata servers.

Greg Ganger, PDL alum Michael Abd-El-Malek (Google), and Bill Courtright enjoy social time at the PDL Retreat.

DISSERTATION ABSTRACT: Enabling Data-Driven Optimization of Quality of Experience in Internet Applications

Junchen Jiang
Carnegie Mellon University, SCS
PhD Defense — June 23, 2017

Today's Internet has become an eyeball economy dominated by applications such as video streaming and VoIP. With most applications relying on user engagement to generate revenues, maintaining high user-perceived Quality of Experience (QoE) has become crucial to ensure high user engagement.

For instance, one short buffering interruption leads to 39% less time spent watching videos and causes significant revenue losses for ad-based video sites. Despite increasing expectations for high QoE, existing approaches have limitations to achieve the QoE needed by today's applications. They either require costly re-architecting of the network core, or use suboptimal endpoint-based protocols to react to the dynamic Internet performance based on limited knowledge of the network.

continued on page 15

continued from page 14

Shinya Matsusmoto (HItachi) talks about his company's research on "Risk-aware Data Replication against Widespread Disasters" at the PDL retreat industry poster session.

In this thesis, I present a new approach, which is inspired by the recent success of data-driven approaches in many fields of computing. I will demonstrate that data-driven techniques can improve Internet QoE by utilizing a centralized real-time view of performance across millions of endpoints (clients). I will focus on two fundamental challenges unique to this data-driven approach: the need for expressive models to capture complex factors affecting QoE, and the need for scalable platforms to make real-time decisions with fresh data from geo-distributed clients.

Our solutions address these challenges in practice by integrating several domain-specific insights in networked applications with machine learning algorithms and systems, and achieve better QoE than using many standard machine learning solutions. I will present end-to-end systems that yield substantial QoE improvement and higher user engagement for video streaming and VoIP. Two of my projects, CFA and VIA, have been used in industry by Conviva and Skype, companies that specialize in QoE optimization for video streaming and VoIP, respectively.

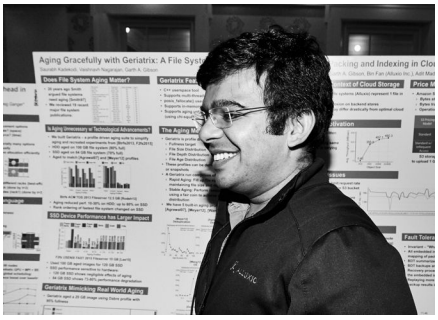
DISSERTATION ABSTRACT: Understanding and Improving the Latency of DRAM-Based Memory System

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 trade-off to improve energy efficiency.

First, while bulk data movement is a key operation in many applications



Saurabh Kadekodi discusses his research on "Aging Gracefully with Geriatrix: A File System Aging Suite" at a PDL retreat poster session.

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.

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 overlapping 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

continued on page 16

continued from page 15

Jiri Schindler (HPE), Bruce Wilson (Broadcom) and Rajat Kateja discuss PDL research at a retreat poster session.

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 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.

THESIS PROPOSAL: Towards Space-Efficient High-Performance In-Memory Search Structures

Huanchen Zhang, SCS
April 30, 2018

This thesis seeks to address the challenge of building space-efficient yet high- performance in-memory search structures, including indexes and filters, to allow more efficient use of memory in OLTP databases. We show that we can achieve this goal by first designing fast static structures that leverage succinct data structures to approach the information-theoretic optimum in space, and then using the "hybrid index" architecture to obtain dynamicity with bounded and modest cost in space and performance.

To obtain space-efficient yet high-performance static data structures, we first introduce the Dynamic-to-Static rules that present a systematic way to convert existing dynamic structures to smaller immutable versions. We then present the Fast Succinct Trie (FST) and its application, the Succinct Range Filter (SuRF), to show how to leverage theories on succinct data structures to build static search structures that consume space close to the information-theoretic minimum while performing comparably to uncompressed indexes. To support dynamic operations such as inserts, deletes, and updates, we introduce the dual-stage hybrid index architecture that preserves the space efficiency brought by a compressed static index, while amortizing its performance overhead on dynamic operations by applying modifications in batches.

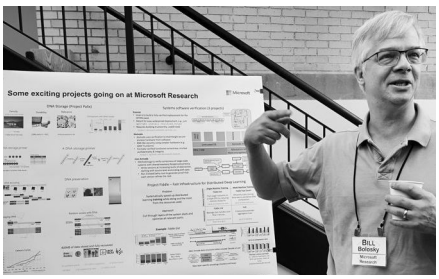
In the proposed work, we seek opportunities to further shrink the size of in-memory indexes by co-designing the indexes with the in-memory tuple storage. We also propose to complete the hybrid index work by extending the techniques to support concurrent indexes.

THESIS PROPOSAL: Efficient Networked Systems for Datacenter Fabrics with RPCs

Anuj Kalia, SCS
March 23, 2018

Datacenter networks have changed radically in recent years. Their bandwidth and latency has improved by orders of magnitude, and advanced network devices such as NICs with Remote Direct Memory Access (RDMA) capabilities and programmable switches have been deployed. The conventional wisdom is that to best use fast datacenter networks, distributed systems must be redesigned to offload processing from server CPUs to network devices. In this dissertation, we show that conventional, non-offloaded designs offer

continued on page 17



Bill Bolosky (Microsoft Research) talks about his company's work on exciting new projects at the PDL retreat industry poster session.

continued from page 16

better or comparable performance for a wide range of datacenter workloads, including key-value stores, distributed transactions, and highly-available replicated services.

We present the following principle: The physical limitations of networks must inform the design of high-performance distributed systems.

Offloaded designs often require more network round trips than conventional CPU-based designs, and therefore have fundamentally higher latency. Since they require more network packets, they also have lower throughput. Realizing the benefits of this principle requires fast networking software for CPUs. To this end, we undertake a detailed exploration of datacenter network capabilities, CPU-NIC interaction over the system bus, and NIC hardware architecture. We use insights from this study to create high-performance remote procedure call implementations for use in distributed systems with active end host CPUs.

We demonstrate the effectiveness of this principle through the design and evaluation of four distributed in-memory systems: a key-value cache, a networked sequencer, an online transaction processing system, and a state machine replication system. We show that our designs often simultaneously outperform the competition in performance, scalability, and simplicity.

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

Joy Arulraj, SCS
December 7, 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.

Joan Digney and Garth Gibson celebrate 25 years of PDL research and retreats.

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.

THESIS PROPOSAL: STRADS: A New Distributed Framework for Scheduled Model-Parallel Machine Learning

Jin Kyu Kim, SCS
May 15, 2017

Machine learning (ML) methods are used to analyze data which are collected from various sources. As the problem size grows, we turn to distributed parallel computation to complete ML training in a reasonable amount of time. However, naive parallelization of ML algorithms often hurts the effectiveness of parameter updates due to the dependency structure among model parameters, and a subset of model parameters often bottlenecks the completion of ML algorithms due to the uneven convergence rate. In this proposal, I propose two efforts: 1) STRADS that improves the training speed in an order of magnitude and 2) STRADS-AP that makes parallel ML programming easier.

In STRADS, I will first present scheduled model-parallel approach with two specific scheduling schemes: 1) model parameter dependency checking to avoid updating dependent parameters concurrently; 2) parameter prioritization to give more update chances to the parameters far from its convergence point. To efficiently run the scheduled model-parallel in a distributed system,

continued on page 18

Yixin Luo and and Michael Kuchnik, ready to discuss their research on "Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid-State Drives" and "Machine Learning Based Feature Tracking in HPC Simulations" at a PDL retreat poster session.

DEFENSES & PROPOSALS

continued from page 17

I implement a prototype framework called STRADS. STRADS improves the parameter update throughput by pipelining iterations and overlapping update computations with network communication for parameter synchronization. With ML scheduling and system optimizations, STRADS improves the ML training time by an order of magnitude. However, these performance gains are at the cost of extra programming burden when writing ML schedules. In STRADS-AP, I will present a high-level programming library and a system infrastructure that automates ML scheduling. The STRADS-AP library consist of three programming constructs: 1) a set of distributed data structures (DDS); 2) a set of functional style operators; and 3) an imperative style loop operator. Once an ML programmer writes an ML program using STRADS-AP library APIs, the STRADS-AP runtime automatically parallelizes the user program over a cluster ensuring data consistency.

Dana Van Aken presents her research on "Automatic Database Management System Tuning Through Large-scale Machine Learning" at the PDL retreat.

THESIS PROPOSAL: Novel Computational Techniques for Mapping Next-Generation Sequencing Reads

Hongyi Xin, SCS
May 31, 2017

DNA read mapping is an important problem in Bioinformatics. With the introduction of next-generation sequencing (NGS) technologies, we are facing an exponential increase in the amount of genomic sequence data. The success of many medical and genetic applications critically depends on computational methods to process the enormous amount of sequence data quickly and accurately. However, due to the repetitive nature of human genome and limitations of the sequencing technology, current read mapping methods still fall short from achieving both high performance and high sensitivity.

In this proposal, I break down the DNA read mapping problem into four subproblems: intelligent seed extraction, efficient filtration of incorrect seed locations, high performance extension and accurate and efficient read cloud mapping. I provide novel computational techniques for each subproblem, including: 1) a novel seed selection algorithm that optimally divides a read into low frequency seeds; 2) a novel SIMD-friendly bit-parallel filtering problem that quickly estimates if two strings are highly similar; 3) a generalization of a state-of-the-art approximate string matching algorithm that measures genetic similarities with more realistic metrics and 4) a novel mapping strategy that utilizes characteristics of a new sequencing technologies, read cloud sequencing, to map NGS reads with higher accuracy and efficiency.

ALUMNI NEWS

Hugo Patterson (Ph.D., ECE '98)

We are pleased to pass on the news that Datrium (www.datrium.com/), where Hugo is a co-founder, won Gold in Search Storage's 2017 Product of the Year.

"Datrium impresses judges and wins top honors with its DVX storage architecture, designed to sidestep latency and deliver performance and speed at scale." http://bit.ly/2Cl2mAR

Hugo received his Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University where he was a charter student in the PDL. He was advised by Garth Gibson and his Ph.D. research focused on informed prefetching and caching. He was named a distinguished alumni of the PDL in 2007.

Ted Wong (Ph.D, CS '04)

Ted joined 23andMe (www.23andme.com), as a Senior Software Engineer with the Machine Learning Engineering group back in January 2018, and reports he is incredibly happy to be there. He also wants to mention that they are hiring!

23andMe is a personal genomics and biotechnology company based in Mountain View, California. The company is named for the 23 pairs of chromosomes in a normal human cell.

Gauri Joshi

The PDL would like to welcome Gauri Joshi to our family! Gauri is an Assistant Professor at CMU in the Department of Electrical and Computer Engineering. She is interested in stochastic modeling and analysis that provides sharp insights into the design of computing systems. Her favorite tools include probability, queueing, coding theory and machine learning.

Until August 2017 Gauri was a Research Staff Member at IBM T. J. Watson in Yorktown Heights NY. In June 2016 she completed her PhD at MIT, working with Prof. Gregory Wornell and Prof. Emina Soljanin. Before that, Gauri spent five years at IIT Bombay, where I completed a dual degree (B.Tech + M.Tech) in Electrical Engineering. She also spent several summers interning at Google, Bell Labs, and Qualcomm.

Currently, Gauri is working on several projects. These include one on Distributed Machine Learning. In large-scale machine learning, training is performed by running stochastic gradient descent (SGD) in a distributed fashion using a central parameter server and multiple servers (learners). Using asynchronous methods to alleviate the problem of stragglers, the research goal is to design a distributed SGD algorithm that strikes the best trade-off between the training time, and errors in the trained model.

Her project on Straggler Replication in Parallel Computing develops insights into the best relaunching time, and the number of replicas to relaunch to reduce latency, without a significant increase in computing costs in jobs with hundreds of parallel tasks, where the slowest task becomes the bottleneck.

Unlike traditional file transfer where only total delay matters, Streaming Communication requires fast and in-order delivery of individual packets to the user. This project analyzes the trade-off between throughput and the in-order delivery delay, and in particular how it is affected by the frequency of feedback to the source, and proposes a simple combination of repetition and greedy linear coding that achieves close to optimal throughput-delay trade-off.

Rashmi Vinayak

We would also like to welcome Rashmi Vinayak! Rashmi is an assistant professor in the Computer Science department at Carnegie Mellon University. She received her PhD in the EECS department at UC Berkeley in 2016, and was a postdoctoral researcher at AMPLab/RISELab and BLISS. Her dissertation received the Eli Jury Award 2016 from the EECS department at UC Berkeley for outstanding achievement in the area of systems, communications, control, or signal processing. Rashmi is the recipient of the IEEE Data Storage Best Paper and Best Student Paper Awards for the years 2011/2012. She is also a recipient of the Facebook Fellowship 2012-13, the Microsoft Research PhD Fellowship 2013-15, and the Google Anita Borg Memorial Scholarship 2015-16. Her research interests lie in building high performance and resource-efficient big data systems based on theoretical foundations.

A recent project has focused on Storage and caching, particularly on fault tolerance, scalability, load balancing, and reducing latency in large-scale distributed data storage and caching systems. She and her colleagues designed coding theory based solutions that were shown to be provably optimal. They also built systems and evaluated them on Facebook's data-analytics cluster and on Amazon EC2 showing significant benefits over the state-of-the-art. The solutions are now a part of Apache Hadoop 3.0 and are also being considered by several companies such as NetApp and Cisco.

Rashmi is also interested in machine learning: the research focus here has been on the generalization performance of a class of learning algorithms that are widely used for ranking. She collaborated on designing an algorithm building on top of Multiple Additive Regression Trees, and through empirical evaluation on real-world datasets showed significant improvement over classification, regression, and ranking tasks. This new algorithm is now deployed in production in Microsoft's data-analysis toolbox which powers the Azure Machine Learning product.

Alex Glikson

Alex Glikson joined the Computer Science Department as a staff engineer, after spending the last 14 years at IBM Research in Israel, where he has been leading a number of research and development projects in the area of systems management and cloud infrastructure. Alex is interested in resource and workload management in cloud computing environments, recently focusing on 'Function-as-a-Service' platforms, infrastructure for Deep Learning workloads, and the combination of the two.

continued from page 7

algorithm [3], and taking tens of data passes to converge, each data pass is slowed down by 30-40% relative to the prior pass, so the eighth data pass is 8.5X slower than the first.

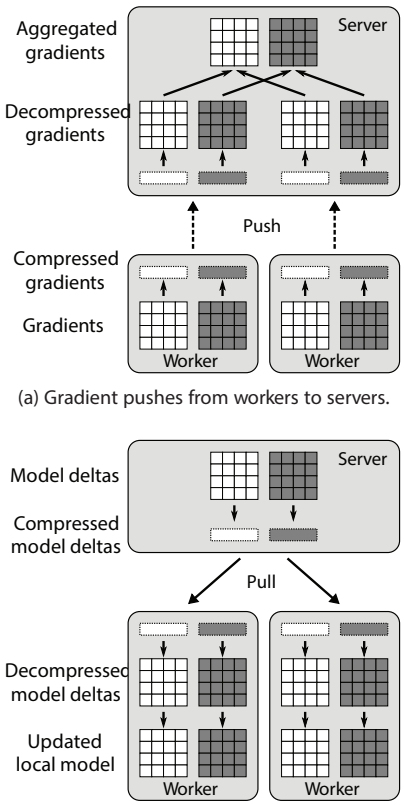
The current practice to avoid such performance penalty is to frequently checkpoint to durable storage device which truncates lineage size. Checkpointing as a performance speedup is difficult for a programmer to anticipate and fundamentally contradicts Spark's philosophy that the working set should stay in memory and not be replicated across the network. Since Spark caches intermediate RDDs, one solution is to cache constructed DAGs and broadcast only new DAG elements. Our experiments show that with this optimization, per iteration execution time is almost independent of growing lineage size and comparable to the execution time provided by optimal checkpointing. On 10 machines using 240 cores in total, without checkpointing we observed a 3.4X speedup when solving matrix factorization and 10X speedup for a streaming application provided in the Spark distribution.

3LC: Lightweight and Effective Traffic Compression for Distributed Machine Learning

Hyeontaek Lim, David G. Andersen & Michael Kaminsky

arXiv:1802.07389v1 [cs.LG] 21 Feb 2018.

The performance and efficiency of distributed machine learning (ML) depends significantly on how long it takes for nodes to exchange state changes. Overly-aggressive attempts to reduce communication often sacrifice final model accuracy and necessitate additional ML techniques to compensate for this loss, limiting their generality. Some attempts to reduce communication incur high computation overhead, which makes their performance benefits visible only over slow networks.



Point-to-point tensor compression for two example layers in 3LC.

We present 3LC, a lossy compression scheme for state change traffic that strikes balance between multiple goals: traffic reduction, accuracy, computation overhead, and generality. It combines three new techniques—3-value quantization with sparsity multiplication, quartic encoding, and zero-run encoding—to leverage strengths of quantization and sparsification techniques and avoid their drawbacks. It achieves a data compression ratio of up to 39–107X, almost the same test accuracy of trained models, and high compression speed. Distributed ML frameworks can employ 3LC without modifications to existing ML algorithms. Our experiments show that 3LC reduces wall-clock training time of ResNet-110–based image classifiers for CIFAR-10 on a 10-GPU cluster by up to 16–23X compared to TensorFlow's baseline design.

LTRF: Enabling High-Capacity Register Files for GPUs via Hardware/Software Cooperative Register Prefetching

Mohammad Sadrosadati, Amirhossein Mirhosseini, Seyed Borna Ehsani, Hamid Sarbazi-Azad, Mario Drumond, Babak Falsafi, Rachata Ausavarungnirun & Onur Mutlu

ASPLOS '18, March 24–28, 2018, Williamsburg, VA, USA.

Graphics Processing Units (GPUs) employ large register files to accommodate all active threads and accelerate context switching. Unfortunately, register files are a scalability bottleneck for future GPUs due to long access latency, high power consumption, and large silicon area provisioning. Prior work proposes hierarchical register file, to reduce the register file power consumption by caching registers in a smaller register file cache. Unfortunately, this approach does not improve register access latency due to the low hit rate in the register file cache.

In this paper, we propose the Latency-Tolerant Register File (LTRF) architecture to achieve low latency in a two-level hierarchical structure while keeping power consumption low. We observe that compile-time interval analysis enables us to divide GPU program execution into intervals with an accurate estimate of a warp's aggregate register working-set within each interval. The key idea of LTRF is to prefetch the estimated register working-set from the main register file to the register file cache under software control, at the beginning of each interval, and overlap the prefetch latency with the execution of other warps. Our experimental results show that LTRF enables high-capacity yet long-latency main GPU register files, paving the way for various optimizations. As an example optimization, we implement the main register file with emerging

continued on page 21

continued from page 20

high-density high-latency memory technologies, enabling 8× larger capacity and improving overall GPU performance by 31% while reducing register file power consumption by 46%.

MASK: Redesigning the GPU Memory Hierarchy to Support Multi-Application Concurrency

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

ASPLOS'18, March 24–28, 2018, Williamsburg, VA, USA.

Graphics Processing Units (GPUs) exploit large amounts of thread-level parallelism to provide high instruction throughput and to efficiently hide long-latency stalls. The resulting high throughput, along with continued programmability improvements, have made GPUs an essential computational resource in many domains. Applications from different domains can have vastly different compute and memory demands on the GPU. In a large-scale computing environment, to efficiently accommodate such wide-ranging demands without leaving GPU resources underutilized, multiple applications can share a single GPU, akin to how multiple applications execute concurrently on a CPU. Multi-application concurrency requires several support mechanisms in both hardware and software. One such key mechanism is virtual memory, which manages and protects the address space of each application. However, modern GPUs lack the extensive support for multi-application concurrency available in CPUs, and as a result suffer from high performance overheads when shared by multiple applications, as we demonstrate.

We perform a detailed analysis of which multi-application concurrency support limitations hurt GPU performance the most. We find that the poor performance is largely a result of the virtual memory mechanisms employed in modern GPUs. In particular, poor address translation performance is a key obstacle to efficient GPU sharing. State-of-the-art address translation mechanisms, which were designed for single-application execution, experience significant inter-application interference when multiple applications spatially share the GPU. This contention leads to frequent misses in the shared translation lookaside buffer (TLB), where a single miss can induce long-latency stalls for hundreds of threads. As a result, the GPU often cannot schedule enough threads to successfully hide the stalls, which diminishes system throughput and becomes a first-order performance concern.

Based on our analysis, we propose MASK, a new GPU framework that provides low-overhead virtual memory support for the concurrent execution of multiple applications. MASK consists of three novel address-translation-aware cache and memory management mechanisms that work together to largely reduce the overhead of address translation: (1) a token-based technique to reduce TLB contention, (2) a bypassing mechanism to improve the effectiveness of cached address translations, and (3) an application-aware memory scheduling scheme to reduce the interference between address translation and data requests. Our evaluations show that MASK restores much of the throughput lost to TLB contention. Relative to a state-of-the-art GPU TLB, MASK improves system throughput by 57.8%, improves IPC throughput by 43.4%, and reduces applicationlevel unfairness by 22.4%. MASK's system throughput is within 23.2% of an ideal GPU system with no address translation overhead.

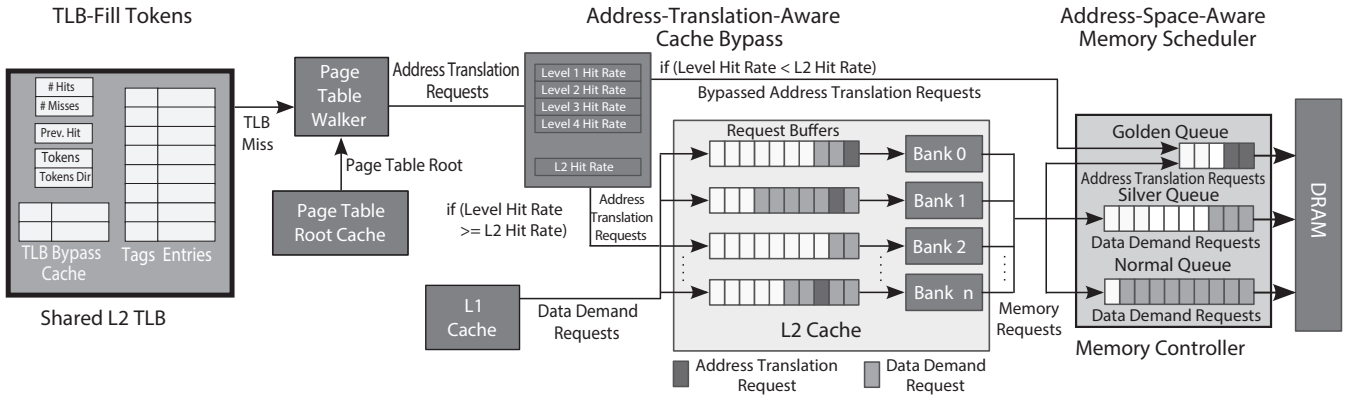
Slim NoC: A Low-Diameter On-Chip Network Topology for High Energy Efficiency and Scalability

Maciej Besta, Syed Minhaj Hassan, Sudhakar Yalamanchili, Rachata Ausavarungnirun, Onur Mutlu & Torsten Hoefler

ASPLOS '18, March 24–28, 2018, Williamsburg, US.

Emerging chips with hundreds and thousands of cores require networks with unprecedented energy/area effi-

continued on page 22



MASK design overview.

continued from page 21

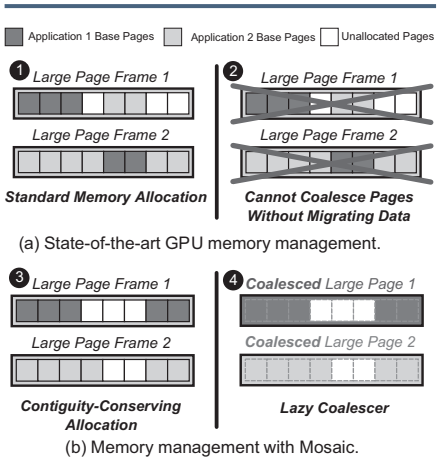
ciency and scalability. To address this, we propose Slim NoC (SN): a new on-chip network design that delivers significant improvements in efficiency and scalability compared to the state-of-the-art. The key idea is to use two concepts from graph and number theory, degree-diameter graphs combined with non-prime finite fields, to enable the smallest number of ports for a given core count. SN is inspired by state-of-the-art off-chip topologies; it identifies and distills their advantages for NoC settings while solving several key issues that lead to significant overheads on-chip. SN provides NoC-specific layouts, which further enhance area/energy efficiency. We show how to augment SN with state-of-the-art router microarchitecture schemes such as Elastic Links, to make the network even more scalable and efficient. Our extensive experimental evaluations show that SN outperforms both traditional low-radix topologies (e.g., meshes and tori) and modern high-radix networks (e.g., various Flattened Butterflies) in area, latency, throughput, and static/dynamic power consumption for both synthetic and real workloads. SN provides a promising direction in scalable and energy-efficient NoC topologies.

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.



Page allocation and coalescing behavior of GPU memory managers: (a) state-of-the-art, (b) Mosaic. 1: The GPU memory manager allocates base pages from both Applications 1 and 2. 2: As a result, the memory manager cannot coalesce the base pages into a large page without first migrating some of the base pages, which would incur a high latency. 3: Mosaic uses Contiguity Conserving Allocation (CoCoA) — a memory allocator which provides a soft guarantee that all of the base pages within the same large page range belong to only a single application, and 4: InPlace Coalescer, a page size selection mechanism that merges base pages into a large page immediately after allocation .

In modern GPUs, we face a trade-off 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 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

continued on page 23

continued from page 22

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.

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.

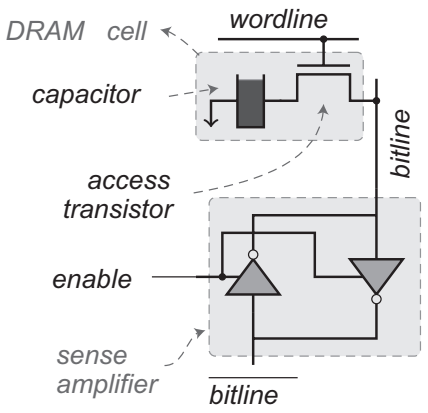
Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology

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.



DRAM cell and sense amplifier.

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

continued on page 24

continued from page 23

state-of-the-art baseline using SIMD optimizations. We describe four other applications that can benefit from Ambit, including a recent technique 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.

Greg opens the 25th PDL Retreat at the Bedford Springs Resort.

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

continued on page 25

continued from page 24

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.

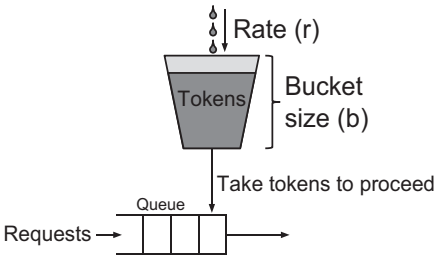
This paper presents an analysis of the differences and similarities between all aforementioned traces. We discuss a variety of aspects: job characteristics, workload heterogeneity, resource utilization, and failure rates. More importantly, we review assumptions from the literature that were originally 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.

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-



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.

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.

This paper describes Workload Compactor, 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, Workload Compactor reduces the number of required servers by 30-60% as compared to state-of-the-art approaches.

Error Characterization, 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.

continued on page 26

continued from page 25

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

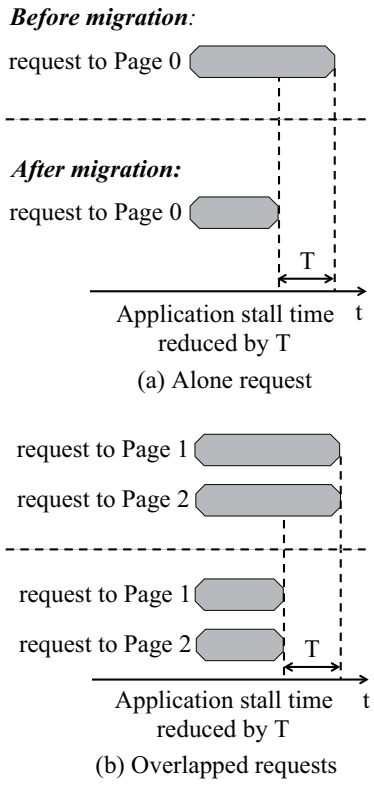
Kristen Gardner, Mor Harchol-Balter, Alan Scheller-Wolf & Benny Van Houdt

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.

Utility-Based Hybrid Memory Management

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



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

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.

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.

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

continued on page 27

continued from page 26

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.

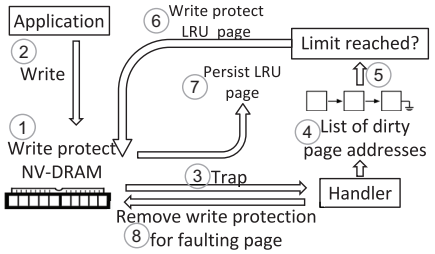
In this paper we use scheduling to address the simultaneous goals of (1) achieving low response time and (2) maintaining fairness across job classes. We develop new exact analysis for per-class response time under First-Come First-Served (FCFS) scheduling for a general type of system structure; our analysis 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.

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



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

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.

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

continued on page 28

continued from page 27

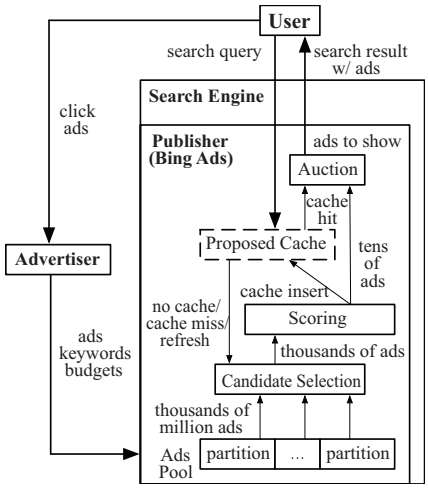
underlying dynamic resource usage of ML applications.

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 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.

continued on page 29

continued from page 28

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.

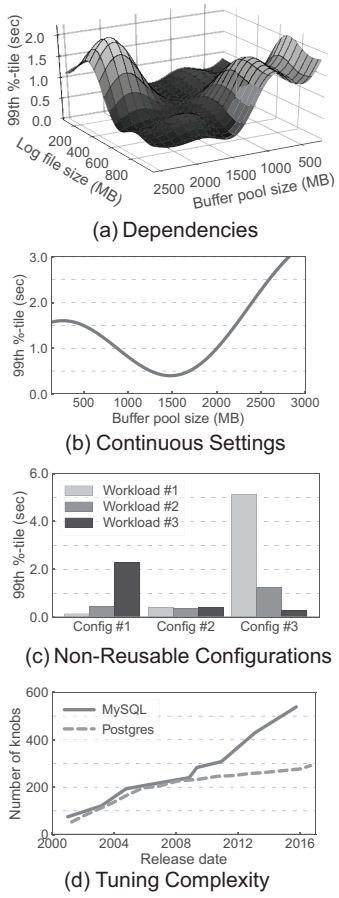
We propose Carpool, the first bufferless on-chip network optimized for one-to-many (i.e., multicast) and many-to-one (i.e., hotspot) traffic. Carpool is based on three key ideas: it (1) adaptively forks multicast flit replicas; (2) merges hotspot flits; and (3) employs a novel parallel port allocation 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.

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.

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

continued on page 30

continued from page 29

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 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.

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.

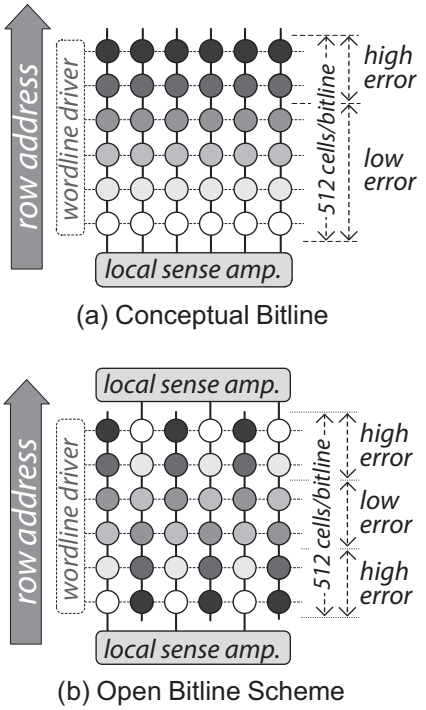
In cloud computing systems, assigning a task to multiple servers and waiting for the earliest copy to finish is an effective method to combat the variability in response time of individual servers and reduce latency. But adding redundancy may result in higher cost of computing resources, as well as an 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 strategy 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.

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



Design-Induced Variation Due to Row Organization

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) 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

continued on page 31

continued from page 30

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.

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.

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.

continued from page 4

- Conglong Li presented "Workload Analysis and Caching Strategies for Search Advertising Systems" at SoCC '17 in Santa Clara, CA.

August 2017

- Kai Ren successfully defended his PhD thesis on "Fast Storage for File System Metadata."
- Souptik Sen interned with LinkedIn's Data group in Sunnyvale, working with Venkatesh Iyer and Subbu Sanka on a data tooling library in Scala which converts generic parameterized Hive queries to Spark to create an optimized workflow on LinkedIn's advertising data pipeline.
- Saurabh Kadekodi interned with Alluxio, Inc. in California, working on packing and indexing in cloud file systems.
- Aaron Harlap interned with Microsoft Research in Seattle, WA, working on "Scaling up Distributed DNN Training."
- Qing Zheng interned with LANL in Los Alamos, NM, working on exascale file systems.
- Charles McGuffey interned with Google in Sunnyvale, CA, working on cache partitioning systems for Google infrastructure.
- Jinliang Wei interned with Saeed Maleki, Madan Musuvathi and Todd Mytkowicz at Microsoft Research in Redmond WA, working on parallelizing and scaling out stochastic gradient descent with sequential semantics.

June 2016

- M. Satyanarayanan and Colleagues Honored for Creation of Andrew File System
- Junchen Jiang successfully defended his PhD dissertation "Enabling Data-Driven Optimization of Quality of Experience in Internet."
- Rajat Kateja presented "Viyojit: Decoupling Battery and DRAM Capacities for Battery-Backed DRAM" at ISCA '17 in Toronto, ON, Canada.
- Utsav Drolia presented "Cachier: Edge-caching for Recognition Applications" at ICDCS '17 in Atlanta, GA.

May 2016

- Hongyi Xin proposed his dissertation research "Novel Computational Techniques for Mapping Next-Generation Sequencing Reads."
- Kevin K. Chang successfully defended his PhD research on "Understanding and Improving the Latency of DRAM-Based Memory System."
- Jin Kyu Kim proposed his PhD research "STRADS: A New Distributed Framework for Scheduled Model-Parallel Machine Learning."
- Dana Van Aken presented "Automatic Database Management System Tuning Through Large-scale Machine Learning" at ICMD '17 in Chicago, IL.
- 19th annual PDL Spring Visit Day.

2017 PDL Workshop and Retreat.