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

Amazon
Facebook
Google
Hewlett Packard Enterprise
Hitachi, Ltd.
IBM Research
Intel Corporation
Microsoft Research
NetApp, Inc.
Oracle Corporation
Pure Storage
Salesforce
Samsung Semiconductor Inc.
Seagate Technology
Two Sigma
Western Digital

PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • SPRING 2021
http://www.pdl.cmu.edu/

PDL NEWS & AWARDS

April 2021
Huanchen Zhang Wins Jim Gray Dissertation Award!

Congratulations to Huanchen Zhang, who has been awarded this year's ACM SIGMOD Jim Gray Dissertation Award. The award recognizes excellent research by doctoral candidates in the database field. Huanchen's dissertation studied "Memory-Efficient Search Trees for Database Management Systems" and addressed the challenge of building compact yet fast in-memory search trees to allow more efficient use of memory in data processing systems.

Since graduation Huanchen has spent time at Snowflake as a Postdoctoral Research Fellow and has now joined Tsinghua University as an Assistant Professor. While at CMU, Huanchen was advised by David Andersen and Andy Pavlo.

April 2021
Juncheng Yang and Rashmi Vinayak Receive NDSI Community Award

Congratulations to Juncheng and Rashmi, who received the NSDI 2021 Community Award for their paper "Segcache: A Memory-efficient and Scalable In-memory Key-value Cache for Small Objects."

The conference, held virtually this year, presents the award for the best

paper whose code and/or data set is made publicly available by the final papers deadline. In collaboration with Twitter, Juncheng and Rashmi building the next generation in-memory caching systems. Segcache enables high memory efficiency, high throughput, and excellent scalability, and demonstrates several important optimization techniques for large-scale web services.

April 2021
Nathan Beckmann Awarded Google Research Scholar Grant

Nathan Beckmann, Assistant Professor, CSD and ECE has received a grant through Google's inaugural Research Scholar Program, which will support his research on " Making Data Access Faster and Cheaper with Smarter Flash Caches". The Research Scholar Program aims to support early-career professors who are pursuing research in fields relevant to Google.

-- info from SCS Wednesday Wire, 4/13/21 and research.google/

continued on page 8

FROM THE DIRECTOR'S CHAIR

GREG GANGER

It is difficult to believe that it has been so long since we've been able to get together in-person to interact, share, brainstorm, and collaborate. We miss you, and each other. But, we are excited that things appear to be moving toward allowing us to gather and host again soon. While we continue to use new ways to connect with our sponsors, such as the new PDL Talk Series of remote presentations by both PDL faculty/students and folks from PDL companies, I am excited to say that we are aggressively planning for a PDL Retreat this Fall (2021). We very much hope to see folks there!

It has been a great year for PDL, on the research and student accomplishment fronts, despite the life challenges we have all faced. Indeed, there has been huge progress on many fronts, new projects and collaborations started, and a bunch of awards, open source releases, and top papers published (e.g., check out all of the OSDI, SOCC, and VLDB papers ;)). A highlight has been strong continued interaction with PDL sponsors, including folks who have given guest lectures to PDL's storage systems and cloud classes and co-authored papers with us in the context of research collaborations. I will not try to cover all of the PDL progress across storage systems, database systems, ML-systems, and data processing infrastructure—specifics can be found throughout the newsletter—but I will highlight a few things.

I'll start with database systems, where the deep dive into automation has emerged with great successes. As Andy explained in the first PDL Talk Series talk of 2021, their exploration of whitebox approaches for database tuning has led to creation of a successful service they call OtterTune. They embedded themselves in real environments to study how well whitebox knob tuning works, what roadblocks emerge in practice, and how they can be worked around. It's poised for huge impact. They also continue to work on blackbox approaches, in which the DBMS is designed from the beginning to automatically adapt. The NoisePage DBMS that they have been building to demonstrate and experiment with this approach is emerging strong. And, although not on those topics, PDL students have won two of the last three Jim Gray Dissertation Awards for database research... just amazing!

PDLers continue to explore exciting opportunities created by new storage technologies, new storage interfaces, and our connections to PDL Consortium companies. Examples of new directions include Flash-specialized caching techniques, NVM-specialized redundancy and remote access approaches, ZNS-specialized software designs, and device-adaptive redundancy approaches for distributed storage systems comprised of heterogeneous mixes of storage devices. We are also exploring offloading of computation to processing elements embedded in storage devices, networking components and memory systems. We thank our PDL sponsor companies who have enabled (and collaborated on) each of the research projects mentioned above by allowing us to experiment with real devices, workload traces, and failure logs!

A third pillar of PDL research is large-scale data processing systems, including systems for ML and schedulers for analytics clusters. Here, also, a lot of new contributions can be found. For example, one of our OSDI 2020 papers describes how inter-job dependency information can be uncovered from provenance information and job logs AND then used to improve scheduling in analytics clusters. An upcoming OSDI 2021 paper describes how adaptive scheduling decisions can be made in shared GPU-clusters running DNN-training jobs.

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
Nathan Beckmann
Chuck Cranor
Lorrie Cranor
Christos Faloutsos
Phil Gibbons
Garth Gibson
Mor Harchol-Balter
Gauri Joshi
Todd Mowry
David O'Hallaron
Andy Pavlo
Majd Sakr
M. Satyanarayanan
Dimitrios Skarlatos
Rashmi Vinayak

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

GRADUATE STUDENTS

Daiyaan Arfeen
Nirav Atre
Mohammad Bakhshalipour
Ben Berg
Vilas Bhat
Amirali Boroumand
Sol Boucher
Matt Butrovich
Damla Senol Cali
Mengxin Cao
Dominic Chen
Zhiran Chen
Yae Jee Cho
Andrew Chung
Ziqi Dong
Pratik Fegade
Ziqiang Feng
Graham Gobieski
Zijing Gu
Samarth Gupta
Jin Han
Travis Hance
Ankush Jain
Ellango Jothimurugesan
Saurabh Arun Kadekodi
Daehyeok Kim
Thomas Kim
Arvind Sai Krishnan
Jack Kosaian
Joseph Koshakow
Michael Kuchnik
Tian Li
Wan Shen Lim
Kaige Liu
Elliot Lockerman
Lin Ma
Ankur Mallick
Francisco Maturana
Sara McAllister
Charles McGuffey
Prashanth Menon
Hojin Park
Aurick Qiao
Brian Schwedock
Baljit Singh
Vikramraj Sitpal
Suhas J Subramanya
Minh Truong
Jianyu Wang
Ziqi Wang
Daniel Wong
Tong Xiao
Ricky Xu
Dongsheng Yang
Jason Yang
Zhengzhe Yang
Ling Zhang
Giulio Zhou

UNDERGRADUATE STUDENTS

Jordi Gonzalez
Julian Tutuncu-Macias

FROM THE DIRECTOR'S CHAIR

And a lot of our recent work to simplify, automate, and improve efficiency in big-data ML systems is being combined and expanded in a big new research effort exploring techniques that could be adopted and adapted by the Army's Pittsburgh-homed AI Integration Center (AI2C), as it creates new mechanisms for AI development and use in the Army.

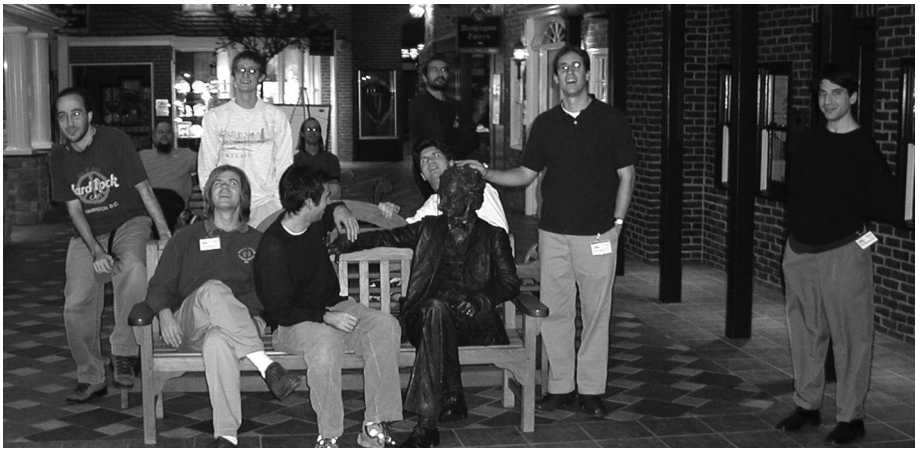
Many other ongoing PDL projects are also producing cool results... too many for me to cover, especially as I strive to keep this note brief. But, this newsletter and the PDL website offer more details and additional research highlights.

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.

PDL Retreat group photo 2001.

Hugo Patterson (PDL 1991-1997, Distinguished Alumni, Network Appliance), John Wilkes (HP) and Greg Ganger.

Jiri Schindler (PDL 1998-2004) and Craig Harmer (Veritas).



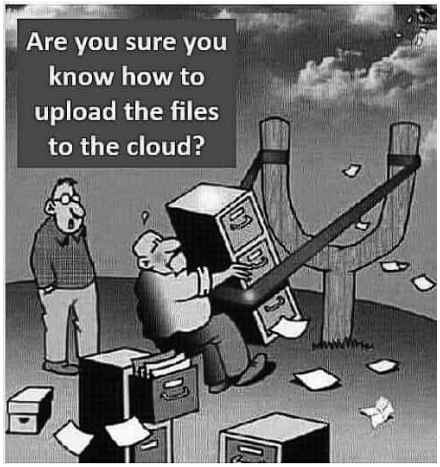
From L-R: Cory Williams, Jay Wylie, John Griffin, Craig Soules, Steve Schlosser, John Strunk, John Bucy, Gregg Economou, Mark Twain, Garth Goodson, David Petrou..

YEAR IN REVIEW (& PHOTOS FROM 20 YEARS AGO)

Since none of us have been able to gather and take pictures for over a year now, we've decided to show PDL life as it was 20 years ago in the newsletter snapshots. Look for them throughout the issue! (Industry guests' affiliation will be listed as it was in 2001).

April 2021

- Huanchen Zhang received the Jim Gray Dissertation Award for his thesis titled "Memory-Efficient Search Trees for Database Management Systems."
- Juncheng Yang and Rashmi Vinayak presented "Segcache: A Memory-efficient and Scalable In-memory Key-value Cache for Small Objects" at NSDI and won the NSDI Community Award.
- Ziqiang Feng successfully defended his PhD research on "Human-efficient Discovery of Edge-based Training Data for Visual Machine Learning" on April 19.
- Daming Chen presented "Hercules: Securing Programs Via Hardware-Enforced Message Queues" at ASPLOS '21.
- Juncheng Yang presented "Segcache: A Memory-Efficient and Scalable In-Memory Key-Value Cache for Small Objects" at the 18th USENIX Symposium on Networked Systems Design and Implementation (NSDI'21) as well as preparing by giving this talk for his speaking skills requirement.
- Giulio Zhou presented "Learning on Distributed Traces for Data Center Storage Systems" at the 4th Conference on Machine Learning and Systems '21.
- Pratik Fegade presented "Cortex: A Compiler for Recursive Deep Learning Models" at the 4th MLSys Conference.
- Prashanth Menon successfully defended his dissertation "On Building Robustness into Compilation-Based Main-Memory Database Query Engines".
- Nathan Beckmann awarded Google Research Scholar grant to further his work on "Making Data Access Faster and Cheaper with Smarter Flash Caches".



March 2021

- Sara McAllister received a 5-year fellowship under the NSF Graduate Research Fellowship Program.
- Greg spoke at the Pittsburgh Technology Council 2021 Beyond Big Data Summit.
- Lorrie Cranor was named as a co-leader of the joint CMU/University of Pittsburgh Collaboratory Against Hate.

February 2021

- Francisco Maturana presented his speaking skills requirement talk on "Convertible Codes: Efficient Conversion of Coded Data in Distributed Storage."

January 2021

- Qing Zheng successfully presented his thesis research on "Distributed Metadata and Streaming Data Indexing as Scalable Filesystem Services."
- Pratik Fegade gave his speaking skills talk on "Scalable Pointer Analysis of Data Structures Using Semantic Models".
- Ling Zhang spoke on "Everything is a Transaction: Unifying Logical Concurrency Control and Physical Data Structure Maintenance in Database Management Systems" at the Conference on Innovative Data Systems Research (CIDR'21).

December 2020

- Dana Van Aken successfully presented her PhD thesis research "On Automatic Database Management System Tuning Using Machine Learning" on December 16.
- Saurabh Kadekodi successfully presented his PhD research on "Disk-Adaptive Redundancy: Tailoring Data Redundancy to Disk-reliability-heterogeneity in Cluster Storage Systems" on December 3.
- Lorrie Cranor was named an American Association for the Advancement of Science (AAAS) Fellow.

November 2020

- Anuj Kalia received the Dennis M. Ritchie Doctoral Dissertation Award Honorable Mention for his research on "Efficient Remote Procedure Calls for Datacenters". Anuj was advised by David Andersen.
- Andrew Chung presented "Unearthing Inter-job Dependencies for Better Cluster Scheduling" at the 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI'20).
- Saurabh Kadekodi gave a presentation on "PACEMAKER: Avoiding HeART Attacks in Storage Clusters with Disk-adaptive Redundancy" at OSDI'20.
- Benjamin Berg spoke on "The CacheLib Caching Engine: Design and Experiences at Scale" at OSDI'20.
- Juncheng Yang discussed "A Large Scale Analysis of Hundreds of In-memory Cache Clusters at Twitter" at OSDI'20.

continued on page 22

Everything is a Transaction: Unifying Logical Concurrency Control and Physical Data Structure Maintenance in Database Management Systems

Ling Zhang, Matthew Butrovich, Tianyu Li, Yash Nannapanei, Andrew Pavlo, John Rollinson, Huanchen Zhang, Ambarish Balakumar, Daniel Biales, Ziqi Dong, Emmanuel Eppinger, Jordi Gonzalez, Wan Shen Lim, Jianqiao Liu, Lin Ma, Prashanth Menon, Soumil Mukherjee, Tanuj Nayak, Amadou Ngom, Jeff Niu, Deepayan Patra, Poojita Raj, Stephanie Wang, Wuwen Wang, Yao Yu, William Zhang

Conference on Innovative Data Systems Research (CIDR) 2021. Virtual Event, January 11-15, 2021.

Almost every database management system (DBMS) supporting transactions created in the last decade implements multi-version concurrency control (MVCC). But these systems rely on physical data structures (e.g., B+trees, hash tables) that do not natively support multi-versioning. As a result, there is a disconnect between the logical semantics of transactions and the DBMS's underlying implementation. System developers must invest in engineering efforts to coordinate transactional access to these data structures and non-transactional maintenance tasks. This burden leads to challenges when reasoning about the system's correctness and performance and inhibits its modularity. In this paper, we propose the Deferred Action Framework (DAF), a new system architecture for scheduling maintenance tasks in an MVCC DBMS integrated with the system's transactional semantics. DAF allows the system to register arbitrary actions and then defer their processing until they are deemed safe by transactional processing. We show that DAF can support garbage collection and index cleaning without compromising performance while facilitating higher-level implementation goals, such as non-blocking schema changes and self-driving optimizations.

Segcache: A Memory-Efficient and Scalable In-Memory Key-Value Cache for Small Objects

Juncheng Yang, Yao Yue, Rashmi Vinayak

18th USENIX Symposium on Networked Systems Design and Implementation (NSDI). Virtual Event, April 12–14, 2021.

Modern web applications heavily rely on in-memory key-value caches to deliver low-latency, high-throughput services. In-memory caches store small objects of size in the range of 10s to 1000s of bytes, and use TTLs widely for data freshness and implicit delete. Current solutions have relatively large per-object metadata and cannot remove expired objects promptly without incurring a high overhead. We present Segcache, which uses a segment-structured design that stores data in fixed-size segments with three key features: (1) it groups objects with similar creation and expiration time into the segments for efficient expiration and eviction, (2) it approximates some and lifts most per-object metadata into the shared segment header and shared information slot in the hash table for object metadata reduction, and (3) it performs segment-level bulk expiration and eviction with tiny critical sections for high scalability. Evaluation using production traces shows that Segcache uses 22-60% less memory than state-of-the-art designs for a variety of workloads. Segcache simultaneously delivers high throughput, up to 40% better than Memcached on a single thread. It exhibits close-to-linear scalability, providing a close to 8× speedup over Memcached with 24 threads.

Learning On Distributed Traces For Data Center Storage Systems

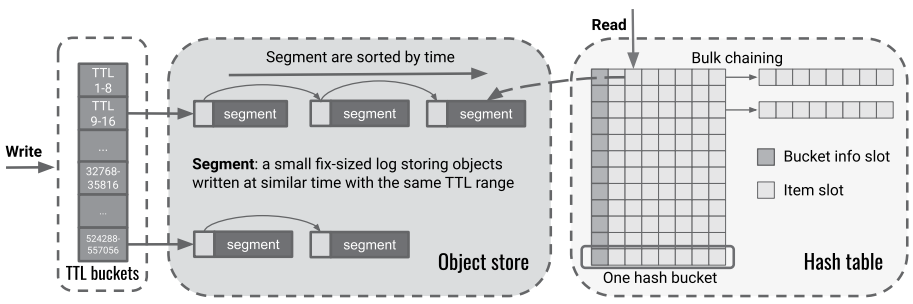
Giulio Zhou, Martin Maas

Conference on Machine Learning and Systems '21, Virtual Event, April 5-9, 2021.

Storage services in data centers continuously make decisions, such as for cache admission, prefetching, and block allocation. These decisions are typically driven by heuristics based on statistical properties like temporal locality or common file sizes. The quality of decisions can be improved through application-level information such as the database operation a request belongs to. While such features can be exploited through application hints (e.g., explicit prefetches), this process requires manual work and is thus only viable for the most tuned workloads.

In this work, we show how to leverage application-level information automatically, by building on distributed traces that are already available in warehouse-scale computers. As these traces are used for diagnostics and

continued on page 6



Overview of Segcache. A read request starts from the hash table (right), a write request starts from the TTL buckets (left).

continued from page 5

accounting, they contain information about requests, including those to storage services. However, this information is mostly unstructured (e.g., arbitrary text) and thus difficult to use. We demonstrate how to do so automatically using machine learning, by applying ideas from natural language processing.

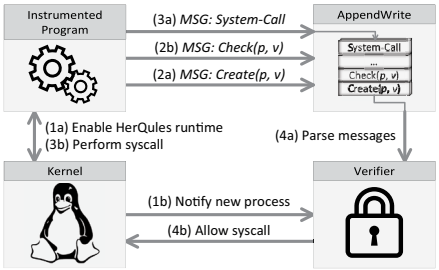
We show that different storage-related decisions can be learned from distributed traces, using models ranging from simple clustering techniques to neural networks. Instead of designing specific models for different storage-related tasks, we show that the same models can be used as building blocks for different tasks. Our models improve prediction accuracy by 11-33% over non-ML baselines, which translates to significantly improving the hit rate of a caching task, as well as improvements to an SSD/HDD tiering task, on production data center storage traces.

Herqules: Securing Programs Via Hardware-Enforced Message Queues

Daming D. Chen, Wen Shih Lim, Mohammad Bakhshalipour, Phillip B. Gibbons, James C. Hoe, Bryan Parno

ASPLOS '21, Virtual Event, April 19–23, 2021.

Many computer programs directly manipulate memory using unsafe pointers, which may introduce memory safety bugs. In response, past work has developed various runtime defenses, including memory safety checks, as well as mitigations like no-execute memory, shadow stacks, and control-flow integrity (CFI), which aim to prevent attackers from obtaining program control. However, software-based designs often need to update in-process runtime metadata to maximize accuracy, which is difficult to do precisely, efficiently, and securely. Hardware-based fine-grained instruction monitoring avoids this problem by maintaining metadata in special-purpose hardware, but suffers from high design complexity and



Overview of interactions under HerQules.

requires significant microarchitectural changes. In this paper, we present an alternative solution by adding a fast hardware-based append-only inter-process communication (IPC) primitive, named AppendWrite, which enables a monitored program to transmit a log of execution events to a verifier running in a different process, relying on inter-process memory protections for isolation. We show how AppendWrite can be implemented using an FPGA or in hardware at very low cost. Using this primitive, we design HerQules (HQ), a framework for automatically enforcing integrity-based execution policies through compiler instrumentation. HerQules reduces overhead on the critical path by decoupling program execution from policy checking via concurrency, without affecting security. We perform a case study on control-flow-integrity against multiple benchmark suites, and demonstrate that HQ-CFI achieves a significant improvement in correctness, effectiveness, and performance compared to prior work.

Cortex: A Compiler for Recursive Deep Learning Models

Pratik Fegade, Tianqi Chen, Phillip B. Gibbons, Todd C. Mowry

4th MLSys Conference, San Jose, CA, Virtual Event, April 2021.

Optimizing deep learning models is generally performed in two steps: (i) high-level graph optimizations such as kernel fusion and (ii) low level kernel optimizations such as those found in vendor libraries. This approach often leaves significant performance on the table, especially for the case of recursive deep learning models. In this paper, we present CORTEX, a compiler-based approach to generate highly-efficient code for recursive models for low latency inference. Our compiler approach and low reliance on vendor libraries enables us to perform end-to-end optimizations, leading to up to 14X lower inference latencies over past work, across different backends.

Unearthing Inter-job Dependencies for Better Cluster Scheduling

Andrew Chung, Subru Krishnan, Konstantinos Karanasos, Carlo Curino, Gregory R. Ganger

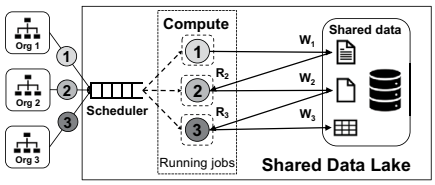
14th USENIX Symposium on Operating Systems Design and Implementation (OSDI'20), Virtual Event, November 4–6, 2020.

Inter-job dependencies pervade shared data analytics infrastructures (so-called "data lakes"), as jobs read output files written by previous jobs, yet are often invisible to current cluster schedulers. Jobs are submitted one-by-one, without indicating dependencies, and the scheduler considers them independently based on priority, fairness, etc. This paper analyzes hidden inter-job dependencies in a 50k+ node analytics cluster at Microsoft, based on job and data provenance logs, finding that nearly 80% of all jobs depend on at least one other job. Yet, even in a business-critical setting, we see jobs that fail because they depend on not-yet-completed jobs, jobs that depend on jobs of lower priority, and other difficulties with hidden inter-job dependencies.

The Wing dependency profiler analyzes job and data provenance logs to find hidden inter-job dependencies, characterizes them, and provides im-

continued on page 7

continued from page 6



Data lake overview: Different jobs submitted by different organizations share the same compute infrastructure and read (R) and write (W) to the same storage system, thereby creating inter-job dependencies as jobs consume the output of other jobs. e.g., Job 2 (from Org 2) reads a file written by Job 1, so Job 2 depends on Job 1.

proved guidance to a cluster scheduler. Specifically, for the 68% of jobs (in the analyzed data lake) that exhibit their dependencies in a recurring fashion, Wing predicts the impact of a pending job on subsequent jobs and user downloads, and uses that information to refine valuation of that job by the scheduler. In simulations driven by real job logs, we find that a traditional YARN scheduler that uses Wing-provided valuations in place of user-specified priorities extracts more value (in terms of successful dependent jobs and user downloads) from a heavily-loaded cluster. By relying completely on Wing for guidance, YARN can achieve nearly 100% of value at constrained cluster capacities, almost 2X that achieved by using the user-provided job priorities.

The CacheLib Caching Engine: Design and Experiences at Scale

Benjamin Berg, Daniel S. Berger, Sara McAllister, Isaac Grosof, Sathya Gunasekar, Jimmy Lu, Michael Uhlar, Jim Carrig, Nathan Beckmann, Mor Harchol-Balter, Gregory R. Ganger

14th USENIX Symposium on Operating Systems Design and Implementation (OSDI'20), Virtual Event, November 4–6, 2020.

Web services rely on caching at nearly every layer of the system architecture. Commonly, each cache is implemented and maintained independently by a distinct team and is highly specialized to its function. For example, an application-data cache would be independent from a CDN cache. However, this approach ignores the difficult challenges that different caching systems have in common, greatly increasing the overall effort required to deploy, maintain, and scale each cache.

This paper presents a different approach to cache development, successfully employed at Facebook, which extracts a core set of common requirements and functionality from otherwise disjoint caching systems. CacheLib is a general-purpose caching engine, designed based on experiences with a range of caching use cases at Facebook, that facilitates the easy development and maintenance of caches. CacheLib was first deployed at Facebook in 2017 and today powers over 70 services including CDN, storage, and application-data caches.

This paper describes our experiences during the transition from independent, specialized caches to the widespread adoption of CacheLib. We explain how the characteristics of production workloads and use cases at Facebook drove important design decisions. We describe how caches at Facebook have evolved over time, including the significant benefits seen from deploying CacheLib. We also discuss the implications our experiences have for future caching design and research.

PACEMAKER: Avoiding HeART Attacks in Storage Clusters with Disk-adaptive Redundancy

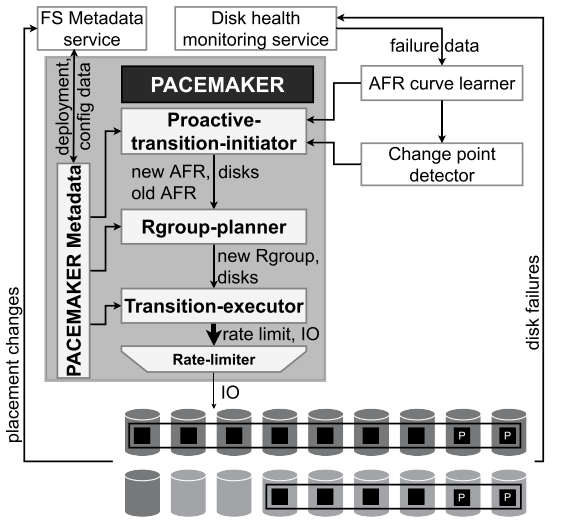
Saurabh Kadekodi, Francisco Maturana, Suhas Jayaram Subramanya, Juncheng Yang, K. V. Rashmi, Gregory R. Ganger

14th USENIX Symposium on Operating Systems Design and Implementation (OSDI'20), Virtual Event, November 4–6, 2020.

Data redundancy provides resilience in large-scale storage clusters, but imposes significant cost overhead. Substantial space-savings can be realized by tuning redundancy schemes to observed disk failure rates. However, prior design proposals for such tuning are unusable in real-world clusters, because the IO load of transitions between schemes overwhelms the storage infrastructure (termed transition overload).

This paper analyzes traces for millions of disks from production systems at Google, NetApp, and Backblaze to expose and understand transition overload as a roadblock to disk-adaptive redundancy: transition IO under existing approaches can consume 100% cluster IO continuously for several weeks. Building on the insights drawn, we present PACEMAKER, a low-overhead disk-adaptive redundancy orchestrator. PACEMAKER mitigates transition overload by (1) proactively organizing data layouts to make future transitions efficient, and (2) initiating transitions proactively in a manner that avoids urgency while not compromising on space-savings. Evaluation of PACEMAKER with traces from four

continued on page 15



PACEMAKER architecture.

continued from page 1

March 2021
Sara McCallister Receives 5-year Graduate Fellowship

Congratulations to Sara, as she receives a 5-year graduate fellowship under the NSF Graduate Research Fellowship Program. The NSF GRFP recognizes and supports outstanding graduate students in NSF-supported STEM disciplines who are pursuing research-based master's and doctoral degrees at accredited US institutions. The five-year fellowship includes three years of financial support including an annual stipend and a cost of education allowance. Sara will be pursuing her Ph.D. in Computer Science in the field of Computer Systems and Embedded Systems.

-- with info from nsfgrfp.org

December 2020
Lorrie Cranor Named AAAS Fellow

Lorrie Cranor, the director of CyLab and a professor in the Institute for Software Research and the department of Engineering and Public Policy, has been named a Fellow of the American Association for the Advancement of Science (AAAS). As part of the section on information, computing, and communication, Cranor was elected as an AAAS Fellow for her contributions to usable privacy and security research, policy, and education.

Election as an AAAS Fellow is an honor bestowed on AAAS members by their peers and has been a tradition since 1874.

-- info from engineering.cmu.edu/news by D. Tkacik

November 2020
Anuj Kalia Receives Dennis M. Ritchie Doctoral Dissertation Award Honorable Mention

Anuj Kalia was awarded honorable mention by the 2020 ACM SIGOPS Dennis M. Ritchie Doctoral Dissertation Award committee for his work on Efficient Remote Procedure Calls for Datacenters. The thesis provides fundamentally grounded guidance about when and where we should split functionality between the CPUs and NICs in [the post-Moore's-law era]. The dissertation takes the approach that datacenter round-trips, measured in microseconds, will only grow increasingly more costly. It then asks how to use modern hardware options to efficiently get that processing to the CPUs nearest the data in ways that meet the requirements for real deployments and avoid both conventional and unexpected sources of overhead.

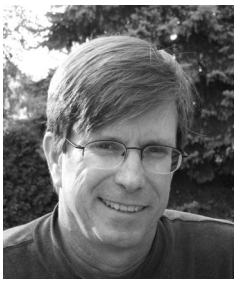
The Dennis M. Ritchie Doctoral Dissertation Award was created in 2013 by ACM SIGOPS to recognize research in software systems and to encourage the creativity that Dennis Ritchie embodied, providing a reminder of Ritchie's legacy and what a difference one person can make in the field of software systems research.

--www.sigops.org/2020/drm-award-2020/

October 2020
PDL Alum Anuj Kalia, and Co-authors David Andersen and Michael Kaminsky Win Best Paper at SoCC'20!

Congratulations to Anuj Kalia, Dave Andersen and Michael Kaminsky, on receiving the best paper award at SoCC'20, which was held virtually this year, for their paper "Challenges and Solutions for Fast Remote Persistent Memory Access." The paper explores the unique challenges that arise when building high-performance networked systems for NVMM.

July 2020
David O'Hallaron Awarded the Philip L. Dowd Fellowship



Congratulations to ECE and CS Professor David O'Hallaron, who has been awarded the Philip L Dowd Fellowship in the College of Engineering. The fellowship is awarded to recognize educational contributions and encourage the undertaking of an educational project such as textbook writing, educational technology development, laboratory experience improvement, educational software, or course and curriculum development. The Dowd Fellowship Award usually consists of a memento and a discretionary fund to support the nominee's education project and lasts for one year beginning the January following the award.

June 2020
Best Paper at SIGMETRICS'20!

Congratulations to Ankur Mallick, Malhar Chaudhari, Ganesh Palanikumar, Utsav Sheth, and Gauri Joshi on receiving the best paper award at the Association for Computing Machinery's (ACM) annual SIGMETRICS conference, which was held virtually in Boston, MA, June 8-12. Their paper, "Rateless Codes for Near-Perfect Load Balancing in Distributed Matrix-Vector Multiplication," proposes a rateless fountain coding strategy that its latency is asymptotically equal to ideal load balancing, and it performs asymptotically zero redundant computation.

NEW PDL FACULTY

Dimitrios Skarlatos

We would like to welcome Dimitrios to CMU and the PDL! Dimitrios starts as an Assistant Professor in the Computer Science Department and Electrical and Computer Engineering (by courtesy) in time for the fall 2021 semester. Currently he is with Facebook Research in Palo Alto.

Dimitrios earned a PhD in Computer Science at the University of Illinois at Urbana-Champaign where he worked with Prof. Josep Torrellas in the i-acoma group. His alma mater is the Technical University of Crete in Greece, where he studied Electronic and Computer Engineering working with Profs. Dionisios Pnevmatikatos, Apostolos Dollas, Ioannis Papaefstathiou and Polyvios Pratikakis.

Dimitrios' research bridges computer architecture and operating systems, focusing on performance, security, and scalability. His current work follows two central themes: uncovering security vulnerabilities and building defenses at the boundary between hardware and OS, and re-designing abstractions and interfaces between the two layers to improve performance and scalability. Recent publications, BabelFish, on architectural support for containers, and Elastic Cuckoo Page Tables were selected as one of 12 IEEE MICRO Top Picks and an IEEE MICRO Top Pick honorable mention, respectively.

"Life's too short for slow and insecure computers!"

PDL ALUMNI NEWS

Abutalib Aghayev (PDL 2015 - 2020) started at Penn State in the Electrical Engineering and Computer Science department last summer as a tenure-track assistant professor.

Henggang Cui (PDL 2012 - 2017) recently left Uber ATG and started a new job at Motional as a "Principal Research Scientist, Team Lead," working on driverless vehicle technology. He is still in Pittsburgh!

Jon Kliegman (PDL 1997-1999) wrote to tell us that Owen Benjamin Kliegman was born April 9, 2020 and just turned one!

Andy Klosterman (PDL 1998-2010) learned in November 2020 that he qualifies for Italian citizenship through his mother and maternal grandmother so now he's busy collecting documentation to prove his EU citizenship. Andy moved from NetApp's Cloud Native Data team (that is spearheading NetApp's Project Astra for Kubernetes Application Management) to being a Product Manager with NetApp's Cloud Insights SaaS product in August 2020.

Michelle Mazurek (PDL 2008-2014) was promoted to Associate Professor (with tenure) last August at the University of Maryland and in January she was named the co-director of the Maryland Cybersecurity Center.

Raja Sambasivan (PDL 2006 - 2016) started as an assistant professor at Tufts University in Medford, MA, in the Fall of 2019. He had one semester on campus before the pandemic struck and has been working from home ever since. Finally, at the beginning of April, he managed to meet his PhD students in person for the second time ever. (The first was when they came to the grad school open house, just before the pandemic struck.)

Niraj Tolia's (PDL 2002 - 2007) startup Kasten was acquired by Veeam last October and is still running as an independent business unit within Veeam. Niraj serves as the GM and President of the group as they plan to triple the size of the team. **Julio Lopez** (PDL 2000 - 2011), another PDL alumni, is also a founding member of Kasten.

Ted Wong (PDL 1997 - 2003) recently became a Tech Lead at 23andMe for the Platforms Engineering team, which designs and builds some of the machine learning analysis pipelines used to study customer data and develop health reports. Ted has been at 23andMe since 2018, and has enjoyed every minute of the challenges developing systems to work with one of the largest genetics databases in the world. During COVID times, he's doubled as a bass singer and the audio engineer for the 23andMe Chromotones a cappella group(!), and taken up playing Fortnite with his kids.

DISSERTATION ABSTRACT: On Building Robustness into Compilation-Based Main-Memory Database Query Engines

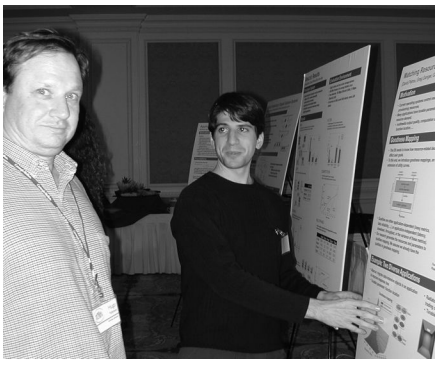
Prashanth Menon
Carnegie Mellon University, SCS

PhD Defense — April 26, 2021

Relational database management systems (DBMS) are the bedrock upon which modern data processing intensive applications are assembled. Critical to ensuring low-latency queries is the efficiency of the DBMSs query processor. Just-in-time (JIT) query compilation is a popular technique to improve analytical query processing performance. However, a compiled query cannot overcome poor choices made by the DBMSs optimizer. A lousy query plan results in lousy query code. Poor query plans often arise and for many reasons. Although there is a large body of work exploring how a query processor can adapt itself at runtime to compensate for inadequate plans, these techniques do not work in DBMSs that rely on compiling queries.

This dissertation presents multiple effective, practical, and complementary techniques to build adaptive query processing into compilation-based engines with negligible overhead. First, we propose a method that intelligently blends two otherwise disparate query processing approaches (compilation and vectorization) into one engine. This necessary first step allows operators to optimize themselves using a combination of software memory prefetching and SIMD vectorization resulting in improved performance. Next, we present a framework that builds upon our previous work to allow the DBMS to modify compiled queries without recompiling the plan or generating code speculatively. This technique enables more extensive groups of operators in a query to coordinate their optimization process. Finally, we present a method that decomposes query plans into fragments that can be compiled and executed independently. This not only reduces compilation overhead but enables the DBMS to learn properties about data processed in an earlier phase of the query to hyper-optimize the code it generates for later phases.

Collectively, the techniques proposed in this dissertation enable any compilation-based DBMS to achieve dynamic runtime robustness without succumbing to any of its overheads.



Retreat poster sessions look much the same these days as they did 20 years ago. Hugo Patterson (PDL 1991-1997 and Network Appliance), David Petrou (PDL 1997-2005).

DISSERTATION ABSTRACT: Human-efficient Discovery of Edge-based Training Data for Visual Machine Learning

Ziqiang Feng
Carnegie Mellon University, SCS

PhD Defense — April 19, 2021

Deep learning enables effective computer vision without hand crafting feature extractors. It has great potential if applied to specialized domains such as ecology, military, and medical science. However, the laborious task of creat labeled training sets of rare targets is a major deterrent to achieving its goal. A domain expert's time and attention is precious. We address this problem by designing, implementing, and evaluating Eureka, a system for human-efficient discovery of rare phenomena from unlabeled visual data. Eureka's central idea is interactive content-based search of visual data based on early-discard and machine learning. We first demonstrate its effectiveness for curating training sets of rare objects. By analyzing contributing factors to human efficiency, we identify and evaluate important system-level optimizations that utilize edge computing and intelligent storage. Lastly, we extend Eureka's methodology to the task of discovering temporal events from video data.

DISSERTATION ABSTRACT: Distributed Metadata and Streaming Data Indexing as Scalable Filesystem Services

Qing Zheng
Carnegie Mellon University, SCS

PhD Defense — January 29, 2021

As people build larger and more powerful supercomputers, the sheer size of future machines will bring unprecedented levels of concurrency. For applications that write one file per process, increased concurrency will cause more files to be accessed simultaneously and this requires the metadata information of these files to be managed more efficiently. An important factor preventing existing HPC filesystems from being able to more efficiently absorb filesystem metadata mutations is the continued use of a single, globally consistent filesystem namespace to serve all applications running on a single computing environment. Having a shared filesystem namespace accessible from anywhere in a computing environment has many welcome benefits, but it increases each application process' communication with the filesystem's metadata servers for ordering concurrent filesystem metadata changes. This is especially the case when all the metadata synchronization and serialization work is coordinated by a small, fixed

continued on page 11

continued from page 10

set of filesystem metadata servers as we see in many HPC platforms today. Since scientific applications are typically self-coordinated batch programs, the first theme of this thesis is about taking advantage of knowledge about the system and scientific applications to drastically reduce, and in extreme cases, remove unnecessary filesystem metadata synchronization and serialization, enabling HPC applications to better enjoy the increasing level of concurrency in future HPC platforms.

Overcoming filesystem metadata bottlenecks during simulation I/O is important. Achieving efficient analysis of large-scale simulation output is an even more important enabler for fast scientific discovery. With future machines, simulations' output will only become larger and more detailed than it is today. To prevent analysis queries from experiencing excessive I/O delays, the simulation's output must be carefully reorganized for efficient retrieval. Data reorganization is necessary because simulation output is not always written in the optimal order for analysis queries. Data reorganization can be prohibitively time-consuming when its process requires data to be readback from storage in large volumes. The second theme of this thesis is about leveraging idle CPU cycles on the compute nodes of an application to perform data reorganization and indexing, enabling data to be transformed to a read-optimized format without undergoing expensive readbacks.

DISSERTATION ABSTRACT: On Automatic Database Management System Tuning Using Machine Learning

Dana Van Aken
Carnegie Mellon University, SCS

PhD Defense — December 16, 2020

Database management systems (DBMSs) are an essential component of any data-intensive application. But tuning a DBMS to perform well is a notoriously difficult task because they have hundreds of configuration knobs that control aspects of their runtime behavior, such as cache sizes and how frequently to flush data to disk. Getting the right configuration for these knobs is hard because they are not standardized (i.e., sets of knobs for different DBMSs vary), not independent (i.e., changing one knob may alter the effects of others), and not uniform (i.e., the optimal configuration depends on the target workload and hardware). Furthermore, as databases grow in both size and complexity, optimizing a DBMS to meet the needs of new applications has surpassed the abilities of even the best human experts. Recent studies using machine learning to automatically configure a DBMS's knobs have shown that such techniques can produce high-quality configurations; however, they need a large amount of training data to achieve good results. Collecting this data is costly and time-consuming.

In this thesis, we seek to address the challenge of developing effective yet practical techniques for the automatic configuration of DBMSs using machine learning. We show that leveraging knowledge gained from previous tuning efforts to assist in the tuning of others can significantly reduce the amount of time and resources needed to tune a DBMS for a new application.

DISSERTATION ABSTRACT: Disk-Adaptive Redundancy: Tailoring Data Redundancy to Disk-reliability-heterogeneity in Cluster Storage Systems

Saurabh Kadekodi
Carnegie Mellon University, SCS

PhD Defense — December 3, 2020

Large-scale cluster storage systems typically consist of a heterogeneous mix of storage devices with significantly varying failure rates. Despite having reliability differences of over 10x in the same storage cluster for the same storage tier, redundancy settings are generally configured in a one-scheme-for-all fashion. This dissertation paves the way for exploiting disk-reliability heterogeneity to tailor redundancy settings to different disk groups for cost-effective, and safer redundancy.

The first contribution is the heterogeneity-aware redundancy tuner (HeART), an online tuning tool that actively engages with the disk hazard (bathtub) curve and identifies the boundaries, and steady-state failure rate for each deployed disk group by make/model. Using this information, HeART suggests the most space-efficient redundancy option allowed that will achieve the specified target data reliability. HeART was evaluated via simulation on a 100,000+ disk production storage cluster where it met target reliability levels while requiring much fewer disks (11–33%) than traditional approaches.

Despite substantial space-savings, HeART is rendered unusable in certain real-world clusters, because the IO load of redundancy transitions overwhelms the storage infrastructure (termed transition overload). The second contribution of this dissertation is an in-depth analysis of millions of disks from Google, NetApp, and Backblaze to understand transition overload as a roadblock for disk-adaptive redundancy. Building on insights drawn from this analysis, Pacemaker is the third contribution of this dissertation; a low-overhead disk-adaptive redundancy orchestrator that mitigates transition overload by initiating transitions proactively and efficiently in a manner that avoids urgency while ensuring high space-savings. Simulation of Pacemaker on four large (110K–450K disks) production clusters shows that the transition IO requirement decreases to <5% cluster IO bandwidth (<0.5% on average). Pacemaker achieves this while providing overall space-savings of 14–20% while never leaving data under-protected.

The final contribution is the design and implementation of disk-adaptive redundancy techniques from Pacemaker in the widely used HDFS. This prototype repurposes HDFS's existing architectural com-

continued on page 12

continued from page 11

ponents for disk-adaptive redundancy, and successfully leverages the robustness of the existing code. The repurposed components are fundamental to any distributed storage system's architecture allowing this prototype to also serve as a guideline for other systems to incorporate disk-adaptive redundancy.

DISSERTATION ABSTRACT: Practical Mechanisms for Reducing Processor-Memory Data Movement in Modern Workloads

Amirali Boroumand
Carnegie Mellon University, ECE

PhD Defense — November 17, 2020

Data movement between the memory system and computation units is one of the most critical challenges in designing high performance and energy-efficient computing system. The high cost of data movement is forcing architects to rethink the fundamental design of computer systems. Recent advances in memory design enable the opportunity for architects to avoid unnecessary data movement by performing Processing-In-Memory (PIM), also known as Near-Data Processing (NDP). While PIM can allow many data-intensive applications to avoid moving data from memory to the CPU, it introduces new challenges for system architects and programmers. Our goal in this thesis is to make PIM effective and practical in conventional computing systems. Toward this end, this thesis presents three major directions: (1) examining the suitability of PIM across key workloads, (2) addressing major system challenges for adopting PIM in computing systems, and (3) re-designing applications aware of PIM capability. In line with these three major directions, we propose a series of practical mechanisms to reduce processor-memory data movement in modern workloads.

First, we comprehensively analyze the energy and performance impact of data movement for several widely-used Google consumer workloads. We find that PIM can significantly reduce data movement for all of these workloads, by performing part of the computation close to memory. Second, we address the coherence challenge for PIM which is one of the major system challenges for adopting PIM in computing systems. We propose CoNDA, a coherence mechanism that lets PIM optimistically execute a PIM kernel, under the assumption that the PIM has all necessary coherence permissions. We show that our proposed mechanism significantly improves performance and reduces energy consumption compared to prior coherence mechanisms. Third, we propose Mensa, a hardware-software co-design approach aware of PIM for Google edge neural network models to enable energy efficient and high performance inference execution. We show that Mensa significantly improves inference energy and throughput, while reducing hardware cost and improving area efficiency over a state-of-the-art edge ML accelerator. Finally, we propose Polynesia, a hardware-software co-designed system aware of PIM for in-memory hybrid transactional/analytical databases. We show that Polynesia significantly outperforms three state-of-the-art HTAP systems and reduces energy consumption

We conclude that the proposed mechanisms by this dissertation provide promising solutions to make PIM more effective and practical in computing systems.

DISSERTATION ABSTRACT: An Approach for Scaling Large-scale Three-dimensional FFT-based Approximate Convolutions on GPUs

Anuva Kulkarni
Carnegie Mellon University, ECE

PhD Defense — August 24, 2020

Supercomputers are needed to process massive datasets in parallel and to execute scientific simulations. Many large-scale scientific simulations are differential equation solvers that involve computing large convolutions using parallel programming. The parallel Fast Fourier Transform (FFT) is widely used

Angela Demke Brown (PDL 1997-2005), now a professor at the University of Toronto, gives her retreat research talk on "Taming the Memory Hogs".

for performing the convolution since it improves computational complexity from O(N2) to O(N log N). However, parallel FFTs require all-to-all communication, which creates data movement bottlenecks, leading to hampered performance and limited scalability. It is a well-known fact that the gap between compute speeds and memory speeds is increasing, leading to performance bottlenecks when the amount of time spent in moving data from memory is more than the time spent in computing on that data. Hence, even though the fastest supercomputers in the world are powered by thousands of Graphical Processing Units (GPUs), their tremendous computing power fails to improve overall performance of the simulation because most of the execution time is spent in communication tasks. Additionally, GPUs have small on-device memory, which means data is transferred into and out of the GPU often during computation, resulting in even more data movement.

This dissertation provides an approach for computing large-scale 3D FFT-based approximate convolutions on heterogeneous computing platforms under certain assumptions. The new approach combines domain decomposition and adaptive sampling to localize the computation

continued on page 13

continued from page 12

and is discussed in the context of a use case in order to highlight the assumptions made regarding properties of the data and convolution kernels. The use case is a Hooke's law partial differential equation solver called MASSIF. Our approach reduces memory footprint of MASSIF and makes processing of large convolutions feasible in the limited memory of a GPU, which in turn allow us to scale the simulation to sizes that were not possible before. An analysis of the algorithm and its trade-offs are presented, along with a theoretical performance model to estimate resource requirements for further scalability.

This work deals with computational problems that lie at the intersection of numerical analysis and signal processing. Differential equation solvers that have similar properties to the use case and that use spectral methods can benefit from the lessons learned in this work. The ability to use GPUs to efficiently perform large 3D convolutions while keeping computation local can benefit many applications such as dislocation dynamics, study of composites, cosmological simulations, and so on.

DISSERTATION ABSTRACT: *Adopting Zoned Storage in Distributed Storage Systems*

Abutalib Aghayev
Carnegie Mellon University, SCS

PhD Defense — August 14, 2020

Hard disk drives and solid-state drives are the workhorses of modern storage systems. For decades, storage systems software has communicated with these drives using the block interface. The block interface was introduced with early hard disk drives, and although it is a poor match for the flash memory used in solid-state drives, it was emulated for backward-compatibility using a translation layer inside drives. More recently, hard disk drives are shifting to shingled magnetic recording, which increases capacity but also violates the block interface. Thus, emerging hard disk drives are also emulating the block interface using a translation layer. These translation layers, however, are hurting the performance and increasing the cost in distributed storage systems.

In this dissertation, we argue for the elimination of the translation layer—and consequently the block interface. We propose adopting the emerging zone interface—a natural fit for both high-capacity hard drives and solid-state drives—and rewriting the storage backend in distributed storage systems to use this new interface. Our thesis is that adopting the zone interface and a special-purpose storage backend—as opposed to using the block interface and a general-purpose file system—will improve the cost-effectiveness and performance of distributed storage systems.

We provide the following evidence to support our thesis. First, we introduce a novel technique to reverse engineer the translation layers of modern hard disk drives and demonstrate their overhead. Second, we reduce the translation layer overhead by optimizing ext4—a general-purpose file system used as a storage backend—on hard drives with a translation layer. While we improve ext4's performance on these drives, we also show that in the presence of a translation layer it is hard to achieve the full potential of a drive with evolutionary changes. Third, we show that general-purpose file systems have high overhead as a storage backend, by studying evolution of storage backends in Ceph—a widely used distributed storage system. Fourth, we adapt the special-purpose storage backend in Ceph to the zone interface. We demonstrate how a special-purpose backend enables quick adoption of the zone interface, and how the zone interface eliminates the translation layer overhead, improving the cost-effectiveness and performance of Ceph.

Simon Towers (HP), Erik Riedel (PDL 1993-1997, Distinguished Alumni, HP) and Garth Gibson at the 2001 PDL Spring Open House.

THESIS PROPOSAL: *Self-Driving Database Management Systems: Forecasting, Modeling, and Planning*

Lin Ma, SCS
November 10, 2020

Database management systems (DBMSs) are an important part of modern data-driven applications. However, they are notoriously difficult to deploy and administer. There are existing methods that recommend physical design or knob configurations for DBMSs. But most of them require humans to make final decisions and decide when to apply changes. Furthermore, they either (1) only focus on a single aspect of the DBMS, (2) are reactionary to the workload patterns and shifts, (3) require expensive exploratory testing on data copies, or (4) do not provide explanations on their decisions/recommendations. Thus, most DBMSs today still require onerous and costly human administration.

In this proposal, we present the design of self-driving DBMSs that enables automatic system management and removes the administration impediments. Our approach consists of three frameworks: (1) workload forecasting, (2) behavior modeling, and (3) action planning. The workload forecasting framework predicts the query arrival rates under varying database workload patterns using an ensemble of time-series forecasting models. The framework also uses a clustering-based technique for reducing the total number of forecasting models to maintain. Our behavior modeling framework constructs and maintains machine learning models that predict the behavior of self-driving DBMS actions: the framework decompos-

continued on page 14

continued from page 13

es a DBMS' architecture into fine-grained operating units to estimate the system's behavior under unseen configurations.

We propose to build the last action planning framework for self-driving DBMSs that make explainable decisions based on the forecasted workload and the modeled behavior. We aim to design a receding horizon control strategy that plans actions using Monte Carlo tree search. We will investigate techniques to reduce the action space and improve the search efficiency to ensure that the planning framework generates the action plan in time. Lastly, we will explore feedback mechanisms to incorporate the observation of the applied actions to correct the planning errors.

THESIS PROPOSAL: Avoiding and Measuring Memory Safety Bugs

Daming Dominic Chen, SCS
October 29, 2020

Many computer programs written in unsafe languages like C and C++ perform low-level memory operations involving pointers, which may accidentally introduce memory safety bugs due to developer error. Common examples of these bugs include buffer overflows, use-after-frees, and double frees, which can all be used by attackers to exploit programs. Indeed, statistics from both Google Chrome and Microsoft have shown that 70% of all security vulnerabilities in their products involve memory safety bugs. Recent research has also demonstrated that these bugs can affect programs written in safe languages like Rust that may contain unsafe code.

Past work has proposed various strategies to detect or mitigate such bugs. These include adding runtime checks (§2.1), randomizing program layout to hide data (§2.2), and validating program execution against models of expected program behavior (§2.3, §2.4). Nevertheless, many of these proposals suffer from high runtime overhead, brittle designs, and/or imprecise analyses, which limits their efficiency and effectiveness. Past work has also developed various methods for measuring the impact of these and other security bugs. Of particular interest are embedded devices, which are widely deployed and occupy a privileged network position, yet are rarely-updated and riddled with security bugs. One approach is internet-scale scanning (§2.5), which requires a feasible network search space, and that remote hosts be both online and remotely-accessible. Another is firmware analysis (§2.6), but which may not accurately model runtime interactions involving multiple programs and scripts.

This thesis addresses these problems as follows: First, we show that emulation can be used to automatically measure the impact of software vulnerabilities in embedded devices. Second, we develop scalable mitigations for memory safety bugs in common software platforms, and quantify our improvements in terms of correctness, effectiveness, and performance.

THESIS PROPOSAL: Co-adaptive Resource Management for Distributed Machine Learning

Aurick Qiao, SCS
September 28, 2020

In the recent decade, machine learning (ML) has found unprecedented success in solving practical problems across diverse application domains, such as recommendation systems, ad-click prediction, sentiment analysis, object detection, and more. Behind this success is an ever-increasing demand for computational resources, which can be leveraged to train larger and more complex models on vaster data. With the availability of hardware resources trending currently towards shared and dynamic computing environments such as clouds and datacenters, efficient and automatic resource management is quickly becoming a key requirement for machine learning in the real world.

Greg Ganger (L) and Natassa Ailamaki (R) getting ready to race to the top of the Nemacolin climbing wall at the 2001 PDL Retreat.

Historically, existing software frameworks which traditionally supported high-performance computing (HPC) or big-data processing workloads, such as MPI and Hadoop, have been re-purposed to additionally support distributed machine learning workloads. More recent frameworks are designed with ML workloads in mind, and have proven to significantly improve ML training time and resource utilization. This thesis proposal takes an evolutionary step along this direction. Furthermore, most ML-oriented resource management systems view the training algorithm as an application-level procedure which should be exactly preserved. We challenge that notion by presenting new systems which deliberately alter their applications during training. Doing so results in better adaptivity to failures, more efficient resource utilization, and automatic configuration of ML applications in dynamic-resource environments.

THESIS PROPOSAL: On Automatic Database Management System Tuning Using Machine Learning

Dana Van Aken, SCS
June 16, 2020

Database management systems (DBMSs) are an important component of any data-intensive application. But tuning a DBMS to perform well is a notoriously difficult task because they have hundreds of configuration knobs that control aspects of their runtime behavior, such as cache sizes and how frequently data is flushed to disk. Getting the right configura-

continued on page 15

continued from page 14

tion for these knobs is hard because they are not standardized (i.e., sets of knobs for different DBMSs vary), not independent (i.e., changing one knob may alter the effects of others), and not universal (i.e., the optimal configuration depends on the target workload and hardware). Furthermore, as databases grow in both size and complexity, optimizing a DBMS to meet the needs of new applications has surpassed the abilities of even the best human experts. Recent studies using machine learning techniques to automatically configure a DBMS's knobs have shown that such techniques are able to produce high-quality configurations, however, they need a large amount of training data to achieve good results. Collecting this data is costly and time-consuming. In this thesis, we seek to address the challenge of developing effective yet practical techniques for the automatic configuration of DBMSs using machine learning. We show that leveraging knowledge gained from previous tuning efforts to assist in the tuning of others can significantly reduce the amount of time and resources needed to tune a DBMS for a new application.

continued from page 7

large (110K–450K disks) production clusters show that the transition IO requirement decreases to never needing more than 5% cluster IO bandwidth (0.2–0.4% on average). PACEMAKER achieves this while providing overall space-savings of 14–20% and never leaving data under-protected. We also describe and experiment with an integration of PACEMAKER into HDFS.

A Large Scale Analysis of Hundreds of In-memory Cache Clusters at Twitter

Juncheng Yang, Yao Yue, K. V. Rashmi

14th USENIX Symposium on Operating Systems Design and Implementation (OSDI'20), Virtual Event, November 4–6, 2020.

Modern web services use in-memory caching extensively to increase throughput and reduce latency. There have been several workload analyses of production systems that have fueled research in improving the effectiveness of in-memory caching systems. However, the coverage is still sparse considering the wide spectrum of industrial cache use cases. In this work, we significantly further the understanding of real-world cache workloads by collecting production traces from 153 in-memory cache clusters at Twitter, sifting through over 80 TB of data, and sometimes interpreting the workloads in the context of the business logic behind them. We perform a comprehensive analysis to characterize cache workloads based on traffic pattern, time-to-live (TTL), popularity distribution, and size distribution. A fine-grained view of different workloads uncover the diversity of use cases: many are far more write-heavy or more skewed than previously shown and some display unique temporal patterns. We also observe that TTL is an important and sometimes defining parameter of cache working sets. Our simulations show that ideal replacement strategy in production caches can be surprising, for example, FIFO works the best for a large number of workloads.

GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework For Genome Sequence Analysis

Damla Senol Cali, Gurpreet S. Kalsi, Zülal Bingöl, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, Onur Mutlu

MICRO'20. 53rd IEEE/ACM International Symposium on Microarchitecture. Virtual Event, Oct 17-21, 2020.

Genome sequence analysis has enabled significant advancements in medical and scientific areas such as personalized medicine, outbreak tracing, and the understanding of evolution. To perform genome sequencing, devices extract small random fragments of an organism's DNA sequence (known as reads). The first step of genome sequence analysis is a computational process known as read mapping. In read mapping, each fragment is matched to its potential location in the reference genome with the goal of identifying the original location of each read in the genome. Unfortunately, rapid genome sequencing is currently bottle-necked by the computational power and memory bandwidth limitations of existing systems, as many of the steps in genome sequence analysis must process a large amount of data. A major contributor to this bottleneck is approximate string matching (ASM), which is used at multiple points during the mapping process. ASM enables read mapping to account for sequencing errors and genetic variations in the reads.

We propose GenASM, the first ASM acceleration framework for genome sequence analysis. GenASM performs bitvector-based ASM, which can efficiently accelerate multiple steps of genome sequence analysis. We modify the underlying ASM algorithm (Bitap) to significantly increase its parallelism

continued on page 16

continued from page 15

and reduce its memory footprint. Using this modified algorithm, we design the first hardware accelerator for Bitap. Our hardware accelerator consists of specialized systolic-array-based compute units and on-chip SRAMs that are designed to match the rate of computation with memory capacity and bandwidth, resulting in an efficient design whose performance scales linearly as we increase the number of compute units working in parallel.

We demonstrate that GenASM provides significant performance and power benefits for three different use cases in genome sequence analysis. First, GenASM accelerates read alignment for both long reads and short reads. For long reads, GenASM outperforms state-of-the-art software and hardware accelerators by 116× and 3.9×, respectively, while reducing power consumption by 37× and 2.7×. For short reads, GenASM outperforms state-of-the-art software and hardware accelerators by 111× and 1.9×. Second, GenASM accelerates pre-alignment filtering for short reads, with 3.7× the performance of a state-of-the-art pre-alignment filter, while reducing power consumption by 1.7× and significantly improving the filtering accuracy. Third, GenASM accelerates edit distance calculation, with 22–12501× and 9.3–400× speedups over the state-of-the-art software library and FPGA-based accelerator, respectively, while reducing power consumption by 548–582× and 67×. We conclude that GenASM is a flexible, high-performance, and low-power framework, and we briefly discuss four other use cases that can benefit from GenASM.

Challenges and Solutions for Fast Remote Persistent Memory Access

Anuj Kalia, David Andersen, Michael Kaminsky

SoCC '20. Virtual Event, October 19–21, 2020. BEST PAPER AWARD!

Non-volatile main memory DIMMs (NVMMs), such as Intel's Optane DC Persistent Memory modules, provide data durability with orders of magnitude higher performance than prior durable technologies. This paper explores the unique challenges that arise when building high-performance networked systems for NVMM. Compared to DRAM, we find that NVMMs have distinctive fundamental properties that pose unique challenges for networked access to NVMM, both from the NIC and the CPU. We show that much of the challenges in efficient access to remote NVMM arises from the fact that CPU caches are not optimized for NVMM. To address these challenges, we propose a menu of solutions for current hardware and evaluate their benefits.

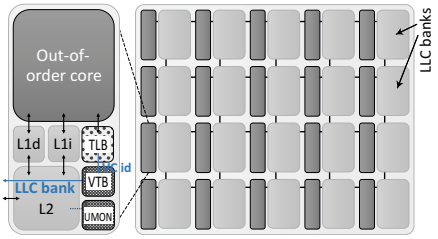
Jumanji: The Case for Dynamic NUCA in the Datacenter

Brian Schwedock, Nathan Beckmann

MICRO '53: Proceedings of the 53nd Annual IEEE/ACM International Symposium on Microarchitecture, Virtual Athens, Greece. Virtual Event, October 17-21, 2020.

The datacenter introduces new challenges for computer systems around tail latency and security. This paper argues that dynamic NUCA techniques are a better solution to these challenges than prior cache designs. We show that dynamic NUCA designs can meet tail-latency deadlines with much less cache space than prior work, and that they also provide a natural defense against cache attacks. Unfortunately, prior dynamic NUCAs have missed these opportunities because they focus exclusively on reducing data movement.

We present Jumanji, a dynamic NUCA technique designed for tail latency and security. We show that prior last-level cache designs are vulnerable to new attacks and offer imperfect performance isolation. Jumanji solves these problems while significantly improving performance of co-running batch applications. Moreover, Jumanji only requires lightweight hardware and a few simple changes to system software, similar to prior D-NUCAs. At 20 cores, Jumanji improves batch weighted speedup by 14% on average, vs. just 2% for a non-NUCA design with weaker security, and is within 2% of an idealized design.



A 20-core system with a distributed LLC (20×1 MB banks). Jumanji adds simple hardware to control data placement, borrowed from Jigsaw. The dotted shape indicates modified components, and the cross-hatched indicates new components.

Permutable Compiled Queries: Dynamically Adapting Compiled Queries without Recompiling

Prashanth Menon, Amadou Ngom, Lin Ma, Todd C. Mowry, Andrew Pavlo

Proceedings of the VLDB Endowment, vol. 14, iss. 2, pages. 101—113, October 2020.

Just-in-time (JIT) query compilation is a technique to improve analytical query performance in database management systems (DBMSs). But the cost of compiling each query can be significant relative to its execution time. This overhead prohibits the DBMS from employing well-known adaptive query processing (AQP) methods to generate a new plan for a query if data distributions do not match the optimizer's estimations. The optimizer could eagerly generate multiple sub-plans for a query, but it can only include a few alternatives as each addition increases the compilation time.

continued on page 17

continued from page 16

We present a method, called Permutable Compiled Queries (PCQ), that bridges the gap between JIT compilation and AQP. It allows the DBMS to modify compiled queries without needing to recompile or including all possible variations before the query starts. With PCQ, the DBMS structures a query's code with indirection layers that enable the DBMS to change the plan even while it is running. We implement PCQ in an in-memory DBMS and compare it against non-adaptive plans in a microbenchmark and against state-of-the-art analytic DBMSs. Our evaluation shows that PCQ outperforms static plans by more than 4× and yields better performance on an analytical benchmark by more than 2× against other DBMSs.

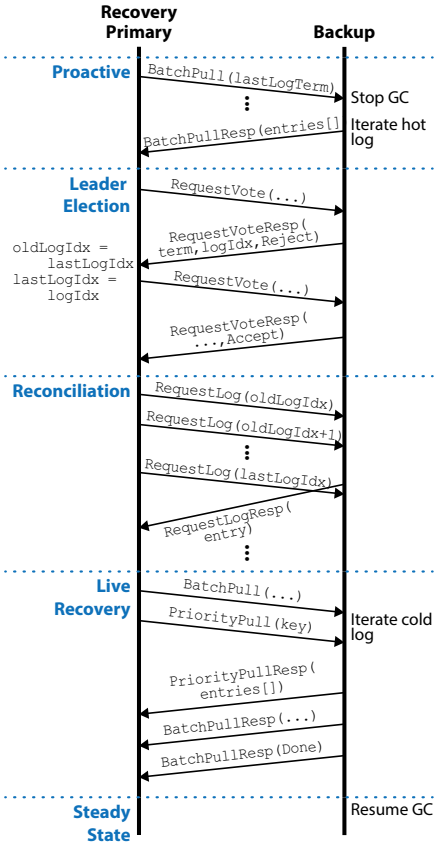
High Availability in Cheap Distributed Key Value Storage

Thomas Kim, Daniel Lin-Kit Wong, Gregory R. Ganger, Michael Kaminsky, David G. Andersen

SoCC '20. Virtual Event, October 19–21, 2020.

Memory-based storage currently offers the highest-performance distributed storage, keeping the primary copy of all data in DRAM. Recent advances in non-volatile main memory (NVMM) technologies promise latency similar to DRAM at reduced cost and energy, but will make providing high availability more challenging. Previous approaches to failure recovery involve maintaining multiple identical replicas or relying on fast offline restoration of data from backup replicas stored on SSD. Unfortunately, NVMM's combination of lower write throughput and increased storage density means that offline restoration can no longer provide sufficiently fast recovery, and maintaining multiple identical replicas is generally cost prohibitive.

CANDStore is a strongly consistent, distributed, replicated key-value store that uses a new fast crash recovery protocol. As a result, CANDStore can use



Recovery timeline.

NVMM and NVMe SSD technology to provide low-latency distributed storage that is cheaper and higher-availability than existing main memory-based distributed storage. Our evaluation shows that CANDStore's recovery protocol enables the system to restore performance and meet SLOs after the failure of a primary node 4.5–10.5x faster than offline recovery.

Accelerating Genome Analysis: A Primer On An Ongoing Journey

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu

An extended and updated version of a paper published in IEEE Micro, vol. 40, no. 5, pp. 65-75, 1 Sept.-Oct. 2020.

Genome analysis fundamentally starts with a process known as read mapping, where sequenced fragments of an organism's genome are compared against a reference genome. Read mapping is currently a major bottleneck in the entire genome analysis pipeline, because state-of-the-art genome sequencing technologies are able to sequence a genome much faster than the computational techniques employed to analyze the genome. We describe the ongoing journey in significantly improving the performance of read mapping. We explain state-of-the-art algorithmic methods and hardware-based acceleration approaches. Algorithmic approaches exploit the structure of the genome as well as the structure of the underlying hardware. Hardware-based acceleration approaches exploit specialized microarchitectures or various execution paradigms (e.g., processing inside or near memory). We conclude with the challenges of adopting these hardware-accelerated read mappers.

Lightweight Preemptible Functions

Sol Boucher, Anuj Kalia, David G. Andersen, Michael Kaminsky

2020 USENIX Annual Technical Conference (USENIX ATC '20). Virtual Boston, MA, July 15–17, 2020.

Lamenting the lack of a natural userland abstraction for preemptive interruption and asynchronous cancellation, we propose lightweight preemptible functions, a mechanism for synchronously performing a function call with a precise timeout that is lightweight, efficient, and composable, all while being portable between programming languages. We present the design of libinger, a library that provides this abstraction, on top of which we build libturquoise, arguably the first general-purpose and backwards-compatible preemptive thread library implemented entirely in userland. Finally, we demonstrate this software stack's applicability to and performance on the problems of

continued on page 18

continued from page 17

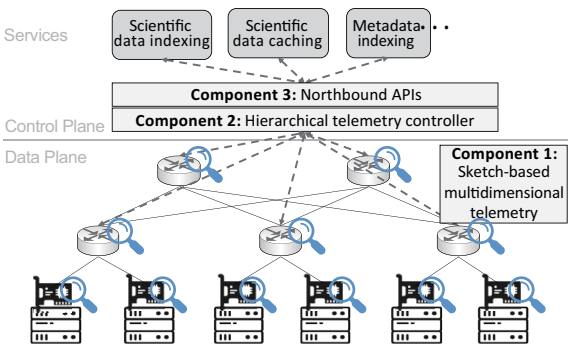
combatting head-offline blocking and time-based DoS attacks.

Unleashing In-Network Computing on Scientific Workloads

Daehyeok Kim, Ankush Jain, Zaoxing Liu, George Amvrosiadis, Damian Hazen, Bradley Settlemyer, Vyas Sekar

arXiv:2009.02457v1 [cs.NI], 5 Sep 2020.

Many recent efforts have shown that in-network computing can benefit various datacenter applications. In this paper, we explore a relatively less-explored domain which we argue can benefit from in-network computing: scientific workloads in high-performance computing. By analyzing canonical examples of HPC applications, we observe unique opportunities and challenges for exploiting in-network computing to accelerate scientific workloads. In particular, we find that the dynamic and demanding nature of scientific workloads is the major obstacle to the adoption of in-network approaches which are mostly open-loop and lack runtime feedback. In this paper, we present NSinC (Network-accelerated ScIeNtific Computing), an architecture for fully unleashing the potential benefits of in-network computing for scientific workloads by providing closed-loop runtime feedback to in-network acceleration services. We outline key challenges in realizing this vision and a preliminary design to enable acceleration for scientific applications.



Overview of NSinC.

Caching with Delayed Hits

Nirav Atre, Justine Sherry, Weina Wang, Daniel S. Berger

SIGCOMM '20, New York, NY. Virtual Event, August 10–14, 2020.

Caches are at the heart of latency-sensitive systems. In this paper, we identify a growing challenge for the design of latency-minimizing caches called delayed hits. Delayed hits occur at high throughput, when multiple requests to the same object queue up before an outstanding cache miss is resolved. This effect increases latencies beyond the predictions of traditional caching models and simulations; in fact, caching algorithms are designed as if delayed hits simply didn't exist. We show that traditional caching strategies – even so called 'optimal' algorithms –can fail to minimize latency in the presence of delayed hits. We design a new, latency-optimal offline caching algorithm called belatedly which reduces average latencies by up to 45% compared to the traditional, hit-rate optimal Belady's algorithm. Using belatedly as our guide, we show that incorporating an object's 'aggregate delay' into online caching heuristics can improve latencies for practical caching systems by up to 40%.We implement a prototype, Minimum-Aggregate-Delay (mad), within a CDN caching node. Using a CDN production trace and backends deployed in different geographic locations, we show that mad can reduce latencies by 12-18% depending on the backend RTTs.

Fast Software Cache Design for Network Appliances

Dong Zhou, Huacheng Yu, Michael Kaminsky, David Andersen

2020 USENIX Annual Technical Conference (USENIX ATC '20). Virtual Event, Boston, MA, July 15–17, 2020.

The high packet rates handled by network appliances and similar software-based packet processing applications place a challenging load on caches such as flow caches. In these environments, both hit rate and cache hit latency are critical to throughput. Much recent work, however, has focused exclusively on one of these two desiderata, missing opportunities to further improve overall system throughput. This paper in- troduces Bounded Linear Probing (BLP), a new cache design optimized for network appliances. BLP works well across different workloads and cache sizes by balancing between hit rate and lookup latency. To accompany BLP, we also present a new, lightweight cache eviction policy called Probabilistic Bubble LRU that achieves near-optimal cache hit rate (assuming the algorithm is offline) without using any extra space. We make three main contributions: a theoretical analysis of BLP, a comparison between existing and proposed cache designs using microbenchmarks, and an end-to-end evaluation of BLP in the popular Open vSwitch (OvS) system. Our end-to-end experiments show that BLP is effective in practice: replacing the microflow cache in OvS with BLP improves throughput by up to 15%.

DriftSurf: A Risk-competitive Learning Algorithm under Concept Drift

Ashraf Tahmasbi, Ellango Jothimurugesan, Srikanta Tirthapura, Phillip B. Gibbons

arXiv:2003.06508 [cs.LG], August, 2020.

When learning from streaming data, a change in the data distribution, also known as concept drift, can render a previously-learned model inaccurate and require training a new model. We present an adaptive learning algorithm

continued on page 19

continued from page 18

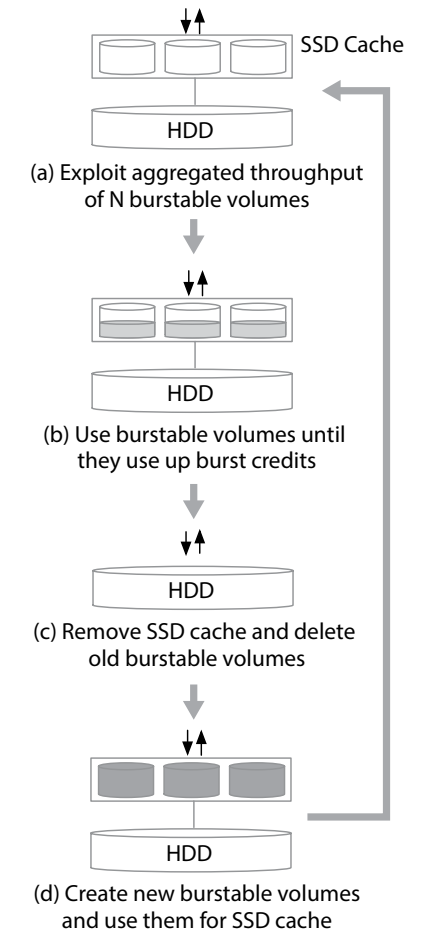
that extends previous drift-detection-based methods by incorporating drift detection into a broader stable-state/reactive-state process. The advantage of our approach is that we can use aggressive drift detection in the stable state to achieve a high detection rate, but mitigate the false positive rate of standalone drift detection via a reactive state that reacts quickly to true drifts while eliminating most false positives. The algorithm is generic in its base learner and can be applied across a variety of supervised learning problems. Our theoretical analysis shows that the risk of the algorithm is competitive to an algorithm with oracle knowledge of when (abrupt) drifts occur. Experiments on synthetic and real datasets with concept drifts confirm our theoretical analysis.

More IOPS for Less: Exploiting Burstable Storage in Public Clouds

Hojin Park, Gregory R. Ganger, George Amvrosiadis

12th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud '20). Virtual Event, Boston, MA, July 13-14, 2020.

Burstable storage is a public cloud feature that enhances cloud storage volumes with credits that can be used to boost performance temporarily. These credits can be exchanged for increased storage throughput, for a short period of time, and are replenished over time. We examine how burstable storage can be leveraged to reduce cost and/or improve performance for three use cases with different data-longevity requirements: traditional persistent storage, caching, and ephemeral storage. Although cloud storage volumes are typically priced by capacity, we find that each AWS gp2 volume starts with the same number of burst credits. Exploiting that fact, we find that aggressive interchanging of large numbers of small short-term volumes can increase IOPS by up to 100 at a cost increase of only 10–40%. Compared to an AWS io1 volume provisioned for the same performance, such interchanging reduces cost by 97.5%.



SSD caching using burstable storage volumes. Initially, (a) N burstable volumes are used as an SSD cache. (b) Cache hits then consume burst credits, and (c) once burst credits are depleted, the volumes are removed and (d) replaced with new burstable volumes.

Order-Preserving Key Compression for In-Memory Search Trees

Huanchen Zhang, Xiaoxuan Liu, David G. Andersen, Michael Kaminsky, Kimberly Keeton, Andrew Pavlo

SIGMOD'20, June 14–19, 2020. Virtual Portland, OR.

We present the High-speed Order-Preserving Encoder (HOPE) for in-memory search trees. HOPE is a fast dictionary-based compressor that encodes arbitrary keys while preserving their order. HOPE's approach is to identify common key patterns at a fine granularity and exploit the entropy to achieve high compression rates with a small dictionary. We first develop a theoretical model to reason about order-preserving dictionary designs. We then select six representative compression schemes using this model and implement them in HOPE. These schemes make different trade-offs between compression rate and encoding speed. We evaluate HOPE on five data structures used in databases: SuRF, ART, HOT, B+tree, and Prefix B+tree. Our experiments show that using HOPE allows the search trees to achieve lower query latency (up to 40% lower) and better memory efficiency (up to 30% smaller) simultaneously for most string key workloads.

Active Learning for ML Enhanced Database Systems

Lin Ma, Bailu Ding, Sudipto Das, Adith Swaminathan

SIGMOD'20, Virtual Event, Portland, OR. June 14–19, 2020.

Recent research has shown promising results by using machine learning (ML) techniques to improve the performance of database systems, e.g., in query optimization or index recommendation. However, in many production deployments, the ML models' performance degrades significantly when the test data diverges from the data used to train these models.

In this paper, we address this performance degradation by using B-instances to collect additional data during deployment. We propose an active data collection platform, ADCP, that employs active learning (AL) to gather relevant data cost-effectively. We develop a novel AL technique, Holistic

continued on page 20

continued from page 19

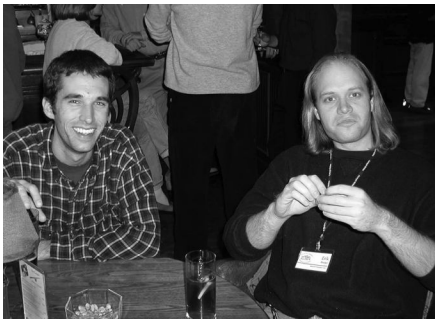
Active Learner (HAL), that robustly combines multiple noisy signals for data gathering in the context of database applications. HAL applies to various ML tasks, budget sizes, cost types, and budgeting interfaces for database applications. We evaluate ADCP on both industry-standard benchmarks and real customer workloads. Our evaluation shows that, compared with other baselines, our technique improves ML models' prediction performance by up to 2× with the same cost budget. In particular, on production workloads, our technique reduces the prediction error of ML models by 75% using about 100 additionally collected queries.

TVARAK: Software-Managed Hardware Offload for Redundancy in Direct-Access NVM Storage

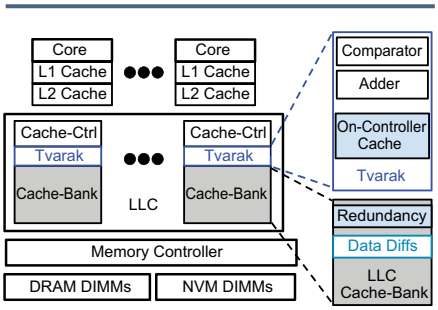
Rajat Kateja, Nathan Beckmann, Greg Ganger

47th International Symposium on Computer Architecture, Virtual Valencia, Spain, May 30 – June 3, 2020.

Production storage systems complement device-level ECC (which covers media errors) with system-checksums and cross-device parity. This system-level redundancy enables systems to detect and recover from data corruption due to device firmware bugs (e.g., reading data from the wrong physical location). Direct access to NVM penalizes software-only implementations



Steve Schlosser (PDL 1998-2004) and Erik Riedel (PDL 1993-1997; HP) enjoying the bar session at the 2001 PDL Retreat.



TVARAK is co-resides with the LLC bank controllers. It includes comparators to identify cache lines that belong to DAX-mapped pages and adders to compute checksums and parity. It includes a small on-controller redundancy cache that is backed by a LLC partition. TVARAK also stores the data diffs to compute checksums and parity.

of system-level redundancy, forcing a choice between lack of data protection or significant performance penalties. We propose to offload the update and verification of system-level redundancy to TVARAK, a new hardware controller co-located with the last-level cache. TVARAK enables efficient protection of data from such bugs in memory controller and NVM DIMM firmware. Simulation-based evaluation with seven data-intensive applications shows that TVARAK is efficient. For example, TVARAK reduces Redis set-only performance by only 3%, compared to 50% reduction for a state-of-the-art software-only approach.

Improving Approximate Nearest Neighbor Search through Learned Adaptive Early Termination

Conglong Li, Minjia Zhang, David G. Andersen, Yuxiong He

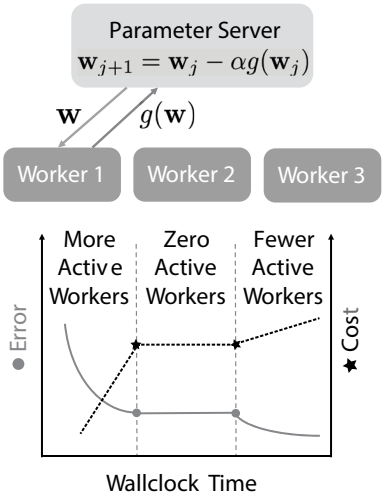
SIGMOD '20. Virtual Event, Portland, OR, June 14–19, 2020.

In applications ranging from image search to recommendation systems, the problem of identifying a set of "similar" real-valued vectors to a query vector plays a critical role. However, retrieving these vectors and computing the corresponding similarity scores from a large database is computationally challenging. Approximate nearest neighbor (ANN) search relaxes the guarantee of exactness for efficiency by vector compression and/or by only searching a subset of database vectors for each query. Searching a larger subset increases both accuracy and latency. State-of-the-art ANN approaches use fixed configurations that apply the same termination condition (the size of subset to search) for all queries, which leads to undesirably high latency when trying to achieve the last few percents of accuracy. We find that due to the index structures and the vector distributions, the number of database vectors that must be searched to find the ground-truth nearest neighbor varies widely among queries. Critically, we further identify that the intermediate search result after a certain amount of search is an important runtime feature that indicates how much more search should be performed.

To achieve a better trade-off between latency and accuracy, we propose a novel approach that adaptively determines search termination conditions for individual queries. To do so, we build and train gradient boosting decision tree models to learn and predict when to stop searching for a certain query. These models enable us to achieve the same accuracy with less total amount of search compared to the fixed configurations. We apply the learned adaptive early termination to state-of-the-art ANN approaches, and evaluate the end-to-end performance on three million to billion-scale datasets. Compared with fixed configurations, our approach consistently improves the average end-to-end latency by up to 7.1 times faster under the same high accuracy targets. Our approach is open source at github.com/efficient/faisslearned-termination.

continued on page 21

continued from page 20



Parameter Server Model and an illustration of how error and cost vary versus training time when the number of workers varies with time. Having more active workers results in a faster decrease in error, but a faster increase in cost.

Machine Learning on Volatile Instances

Xiaoxi Zhang, Jianyu Wang, Gauri Joshi, Carlee Joe-Wong

IEEE Intl. Conf. on Computer Communications (INFOCOM). Virtual Event, Toronto, Canada, July 6-9, 2020.

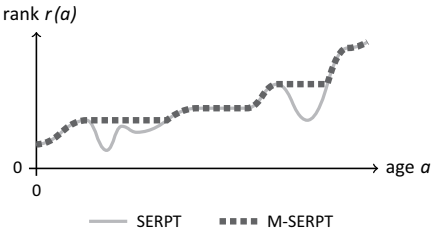
Due to the massive size of the neural network models and training datasets used in machine learning today, it is imperative to distribute stochastic gradient descent (SGD) by splitting up tasks such as gradient evaluation across multiple worker nodes. However, running distributed SGD can be prohibitively expensive because it may require specialized computing resources such as GPUs for extended periods of time. We propose cost-effective strategies that exploit volatile cloud instances that are cheaper than standard instances, but may be interrupted by higher priority workloads. To the best of our knowledge, this work is the first to quantify how variations in the number of active worker nodes (as a result of preemption) affects SGD convergence and the time to train the model. By understanding these trade-offs between preemption probability of the instances, accuracy, and training time, we are able to derive practical strategies for configuring distributed SGD jobs on volatile instances such as Amazon EC2 spot instances and other preemptible cloud instances. Experimental results show that our strategies achieve good training performance at substantially lower cost.

Simple Near-Optimal Scheduling for the M/G/1

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

Proceedings of the ACM Measurement and Analysis of Computer Systems (SIGMETRICS), Virtual Event, Boston, MA, June 2020,

We consider the problem of preemptively scheduling jobs to minimize mean response time of an M/G/1 queue. When we know each job's size, the shortest remaining processing time (SRPT) policy is optimal. Unfortunately, in many settings we do not have access to each job's size. Instead, we know only the job size distribution. In this setting the Gittins policy is known to minimize mean response time, but its complex priority structure can be computationally intractable. A much simpler alternative to Gittins is the shortest expected remaining processing time (SERPT) policy. While SERPT is a natural extension of SRPT to unknown job sizes, it is unknown whether or not SERPT is close to optimal for mean response time.



Example of SERPT and M-SERPT Rank Functions.

We present a new variant of SERPT called monotonic SERPT (M-SERPT) which is as simple as SERPT but has provably near-optimal mean response time at all loads for any job size distribution. Specifically, we prove the mean response time ratio between M-SERPT and Gittins is at most 3 for load $\rho \leq 8/9$ and at most 5 for any load. This makes M-SERPT the only non-Gittins scheduling policy known to have a constant-factor approximation ratio for mean response time.

PDL Consortium Industry guests at the 2001 PDL Retreat. From L to R: John Wilkes (HP), John Howell (Microsoft), Bruce Worthington (Microsoft), John Howard (Sun Microsystems) and Craig Harmer (Veritas).

continued from page 4

- Amirali Boroumand successfully presented his PhD dissertation on "Practical Mechanisms for Reducing Processor-Memory Data Movement in Modern Workloads."
- Lin Ma proposed his thesis research on "Self-Driving Database Management Systems: Forecasting, Modeling, and Planning."

October 2020

- Daming Dominic Chen proposed his thesis research on "Avoiding and Measuring Memory Safety Bugs."
- Anuj Kalia presented "Challenges and Solutions for Fast Remote Persistent Memory Access" at SoCC '20. He and his co-authors David Andersen and Michael Kaminsky won the Best Paper Award!
- Thomas Kim presented "High Availability in Cheap Distributed Key Value Storage" at SoCC '20.
- Damla Senol Cali gave a talk on "GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework For Genome Sequence Analysis" at the 53rd IEEE/ACM International Symposium on Microarchitecture (MICRO'20).
- Brian Schwedock presented "Jumanji: The Case for Dynamic NUCA in the Datacenter" at the 53rd IEEE/ACM International Symposium on Microarchitecture (MICRO'20).
- "Permutable Compiled Queries: Dynamically Adapting Compiled Queries without Recompiling" by Prashanth Menon, Amadou Ngom, Lin Ma, Todd C. Mowry, Andrew Pavlo appeared in Proceedings of the VLDB Endowment, October 2020.

September 2020

- Aurick Qiao proposed his thesis research on "Co-adaptive Resource Management for Distributed Machine Learning."
- "Accelerating Genome Analysis: A Primer On An Ongoing Journey" by Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu appeared in IEEE Micro, September 2020.

August 2020

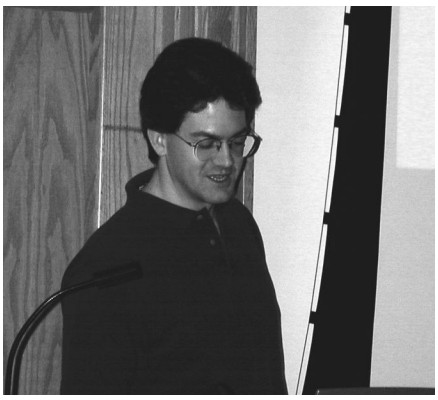
- Michael Kuchnik spent the summer interning at Google researching machine learning data pipelines.
- Anuva Kulkarni successfully defended her PhD research on "An Approach for Scaling Large-scale Three-dimensional FFT-based Approximate Convolutions on GPUs."
- Nirav Atre presented "Caching with Delayed Hits" at SIGCOMM '20.
- Abutalib Aghayev successfully defended his dissertation "Adopting Zoned Storage in Distributed Storage Systems."

July 2020

- David O'Hallaron was awarded the Philip L. Dowd Fellowship.
- Andrew Chung presented his speaking skills talk on "Wing: Unearthing Inter-job Dependencies for Better Cluster Scheduling".
- Aurick Qiao gave his speaking skills talk on "Pollux: Co-adaptive Cluster Scheduling for Goodput-Optimized Deep Learning".
- Dong Zhou presented "Fast Software Cache Design for Network Appliances" at the 2020 USENIX Annual Technical Conference (USENIX ATC '20).
- Hojin Park gave a talk on"More IOPS for Less: Exploiting Burstable Storage in Public Clouds" at the 12th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud '20).
- Xiaoxi Zhang presented "Machine Learning on Volatile Instances," which was co-authored by Jianyu Wang, Gauri Joshi, and Carlee Joe-Wong at the IEEE Intl. Conf. on Computer Communications (INFOCOM).

June 2020

- 22nd Annual/1st Virtual Spring Visit Day
- Ankur Mallick, Malhar Chaudhari, Ganesh Palanikumar, Utsav Sheth, and Gauri Joshi received the best paper award at the Association for Computing Machinery's (ACM) annual SIGMETRICS conference, for their paper, "Rateless Codes for Near-Perfect Load Balancing in Distributed Matrix-Vector Multiplication."
- Rajat Kateja presented "TVARAK: Software-Managed Hardware Offload for Redundancy in Direct-Access NVM Storage" at the 47th International Symposium on Computer Architecture.
- Lin Ma talked about "Active Learning for ML Enhanced Database Systems" at SIGMOD'20.
- Ziv Scully, a student of Mor Harchol-Balter presented "Simple Near-Optimal Scheduling for the M/G/1" at SIGMETRICS 2020.
- Conglong Li presented "Improving Approximate Nearest Neighbor Search through Learned Adaptive Early Termination" at SIGMOD '20, June 14–19, 2020, Virtual Portland, OR, USA.



Greg Ganger welcoming guests to the 2001 PDL retreat!