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
Datadog
Google
Honda
IBM Research
Intel Corporation
Jane Street
Meta
Microsoft Research
Oracle Corporation
Pure Storage
Salesforce
Samsung Semiconductor Inc.
Two Sigma
Western Digital

PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • FALL 2024
http://www.pdl.cmu.edu/

RECENT PUBLICATIONS

Hit the Gym: Accelerating Query Execution to Efficiently Bootstrap Behavior Models for Self-Driving Database Management Systems

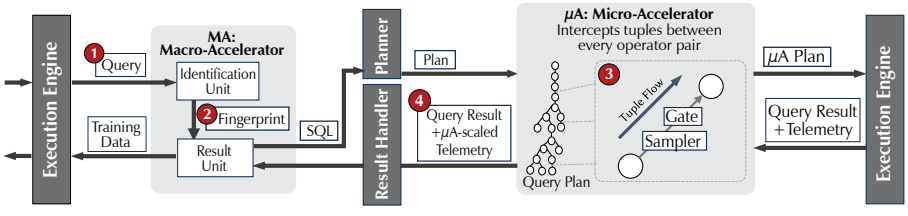
Wan Shen Lim, Lin Ma, William Zhang, Matthew Butrovich, Samuel Arch, Andrew Pavlo

Proceedings of the VLDB Endowment, Vol. 17, No. 11, ISSN 2150-8097. July 2024.

Autonomous database management systems (DBMSs) aim to optimize themselves automatically without human guidance. They rely on machine learning (ML) models that predict their run-time behavior to evaluate whether a candidate configuration is beneficial without the expensive execution of queries. However, the high cost of collecting the training data to build these models makes them impractical for real-world deployments. Furthermore, these models are instance-specific and thus require retraining whenever the DBMS's environment changes. State-of-the-art methods spend over 93% of their time running queries for training versus tuning.

To mitigate this problem, we present the Boot framework for automatically accelerating training data collection in DBMSs. Boot utilizes macro- and micro-acceleration (MMA) techniques that modify query execution semantics with approximate run-time telemetry and skip repetitive parts of the training process. To evaluate Boot, we integrated it into a database gym for PostgreSQL. Our experimental evaluation shows that Boot reduces training collection times by up to 268× with modest degradation in model accuracy. These results also indicate that our MMA-based approach scales with dataset size and workload complexity.

continued on page 5



Architecture – An overview of Boot's internal components and execution flow. The MA component decides whether to execute a query, and the μA component accelerates the execution of a specific query.

FROM THE DIRECTOR'S CHAIR

GREG GANGER

Hello from fabulous Pittsburgh!

It has been another great year for PDL, on the research and student accomplishment fronts, and it was great to get back to a full-scale PDL Retreat while also having a great PDL Talk Series over the summer.

There are exciting new distributed storage and sustainability projects, large numbers of students taking PDL faculty's DB, cloud, and storage systems classes, and lots of great new activities and results in long-standing areas of strength like database systems, storage systems, ML <-> systems, and data processing infrastructure. And, along the way, many students have graduated and joined PDL Consortium companies, PDL researchers have won some big awards, and many cool papers have been published. A great enabler of PDL's success has been strong continued interaction with PDL sponsors, including cool guest lectures in the storage systems, DB, and cloud classes and co-authored papers in the context of research collaborations. Specifics can be found throughout the newsletter, but let me highlight a few things.

I'm really excited about two new inter-related projects focused on bulk storage scalability and data center sustainability. On the sustainability front, PDL researchers have collaborated with Microsoft to identify and quantify carbon footprint contributions of data center components, and then use the results to identify paths for efficiency improvement. One branch of work has designed carbon-efficient compute server designs, referred to as GreenSKUs... check out the ISCA 2024 paper listed on page 7. Another branch exposes the fact that storage systems account for over half of the embodied carbon footprint in data centers, and will be the dominant aspect as green energy usage continues to increase. Key to reducing storage's carbon footprint will be using fewer, larger disks and SSDs, but doing so risks running into an IO bottleneck.

Whether for cost or sustainability, higher-capacity disks are key to reducing the number of devices needed... but each capacity increase comes with lower IO-per-TB-stored since the larger devices don't come with more throughput. Our new declarative IO project, inspired initially by our long-running collaboration with Googlers on adaptive redundancy and efficient transcoding, seeks to mitigate this issue. A lot (often most) of IO in modern bulk storage systems is for data maintenance, like compaction, capacity balancing, integrity checking, etc. Declarative IO interfaces would allow such processes to describe their IO needs so that a new IO planner component can exploit their order- and time-flexibility to eliminate redundant IO (e.g., two reads of the same data in the same day) that caching would not catch. We have great hopes and have been talking in depth about it with many of the PDL companies.

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

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)
ganger@ece.cmu.edu

George Amvrosiadis
David Andersen
Nathan Beckmann
Chuck Cranor
Lorrie Cranor
Christos Faloutsos
Phil Gibbons
Mor Harchol-Balter
Zhihao Jia
Gauri Joshi
Todd Mowry
David O'Hallaron
Jignesh Patel
Andy Pavlo
Majd Sakr
M. Satyanarayanan
Dimitrios Skarlatos
Akshitha Sriraman
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
Yiwei Chen
Joan Digney
Chad Dougherty
Mitch Franzos
Baljit Singh

VISITING RESEARCHERS & POST DOCS

Martin Prammer

GRADUATE STUDENTS

Nikhil Agarwal
Sam Arch
Daiyaan Arfeen
Sanjith Athlur
Nirav Atre
Jennifer Brana
Hilbert Chen
Siyuan Chen
Xinhao Cheng
Val Choung
Patrick Coppock
Theo Gregersen
Ankush Jain
Neharika Jali
Siddharth Jayashankar
Jekyeom Jeon
Sheng Jiang
Hongyi Jin
Hyoungjoo Kim
Timothy Kim
Vasileios Kypriotis
Ruihang Lai
Christos Laspias
Edwin Lim
Wan Shen Lim
Yuchen Liu
Sara McAllister
Yixuan Mei
Deepanjali Mishra
Veronica Muriga
Nj Mukherjee
Gabriele Oliaro
Hojin Park
Ziyue Qiu
Minya Rancic
Hugo Sadok
Sara M Shahri
Eliot Solomon
Shalini Shukla
Suhas J Subramanya
Minh Truong
Jaylen Wang
Patrick Wang
Daniel Wong
Mengdi Wu
Mingkuan Xu
Juncheng Yang
Brian Zhang
Tianyu Zhang
Will Zhang
Kaiyang Zhao
Yiwei Zhao

UNDERGRADUATE STUDENTS

Kyle Booker
Sophia (Qingyang) Cao
Bob (Qinghan) Chen
Benjamin Owad
Helen Wang
Lucy Wang

FROM THE DIRECTOR'S CHAIR

There is an exciting new energy in the database systems research space as well, with great results from the long-running effort to make databases be self-driving... see the VLDB 2024 papers listed on page 1 and page 5, for example. New directions in exploiting BPF for DB visibility and adaptivity as well as very compute-efficient DB structures and algorithms represent exciting new paths forward for the growing database activities. There has also been great progress in our cross-cloud data processing work, including with our Macaron auto-caching approach and a new design for automatically partitioning queries and tables in a hybrid cloud setting (i.e., on-prem plus public cloud).

I've also been very excited to see some of the impactful outcomes from projects that had been ongoing over the past few years. For example, the SIEVE and S3-FIFO cache policies not only won research awards in this past year, they are being adopted at multiple PDL companies. The GPU cluster scheduling research, extended in scale and to heterogeneous GPUs with the Sia scheduler, is also helping turn designs in real environments. We've also been excited to hear that at least one PDL sponsor is now integrating ideas from PDL's disk-adaptive redundancy research into production systems. We thank all of the PDL sponsor companies who have enabled (and collaborated on) much of the research mentioned above by working with us to enable evaluations with real devices, workload traces, and failure logs!

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, staff, and faculty, and it's a pleasure to work with them. As always, their accomplishments point at great things to come.

The PDL was pleased to welcome Niraj Tolia, CTO of Veeam (previously co-founder of Alcion and Kasten), and formerly with Dell, Imaginatics, etc., to the 2023 Retreat and delighted to introduce him as the newest member of the PDL Distinguished Alumni roll.

YEAR IN REVIEW

September 2024

- Sara McAllister named 2025 Siebel Scholar.
- Dimitrios Skarlatos Recognized as an Intel Rising Star.

August 2024

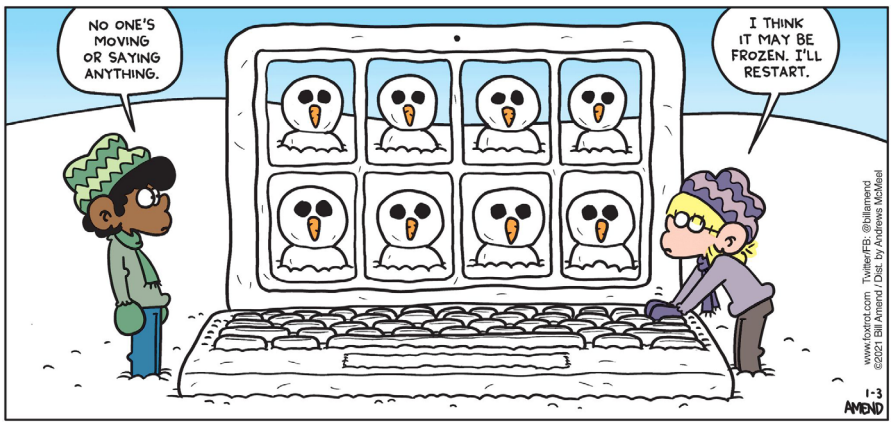
- Daniel Lin-Kit Wong presented his dissertation: "Machine Learning for Flash Caching in Bulk Storage Systems."
- Juncheng Yang defended his dissertation on "Designing Efficient and Scalable Cache Management Systems."
- Suhas Jayaram Subramanya proposed his PhD research on "Efficient Job-resource Co-Adaptivity for Deep learning Workloads on Large Heterogeneous GPU Clusters." He prepared by offering part of his research to the PDL Summer Talk Series: "Sia: Heterogeneity-aware, goodput-optimized ML-cluster Scheduling."
- Travis Hance presented his PhD research "Verifying Concurrent Systems Code."

July 2024

- Akshitha Sriraman received the George Tallman Ladd Research Award.
- Mihailo Rancic won the Hsu Chang Memorial Fellowship, Julia Randall Weertman Award, Phillips and Huang Family Fellowship in Energy, and NSF graduate research fellowship.
- Sara Mahdizadeh Shahri won the K&L Gates Presidential Fellowship, the CMU College of Engineering Presidential Fellowship, and a Boeing Scholarship.
- Jaylen Wang won the Jack and Mildred Bowers scholarship in Engineering, an NSF graduate research fellowship, a Ford foundation pre-doctoral fellowship, and the Benjamin Garver Lamme Westinghouse Graduate fellowship.
- Sara McAllister presented "A Call for Research on Storage Emissions" at HotCarbon'24, Santa Cruz, CA. She also presented "FairyWREN: A Sustainable Cache for Emerging Write-Read-Erase Flash Interfaces" at OSDI '24 in Santa Clara, CA, USA.

June 2024

- Juncheng Yang discussed his work on "Designing Efficient and Scalable Cache Management Systems" in the PDL Summer Talk Series.
- Mohammad Bakshalipour and Phil Gibbons received the Best Paper award for "Agents of Autonomy: A Systematic Study of Robotics on Modern Hardware" at Sigmetrics 2024, held in Venice, Italy this summer.
- Jaylen Wang presented "Designing Cloud Servers for Lower Carbon" at the 51st Intl. Symposium on Computer Architecture (ISCA 2024).
- Akshitha Sriraman presented a keynote talk at the uArch workshop, co-located with ISCA, on "Debunking Myths on What it Takes to do a PhD in CS." She also gave a keynote talk at the CSL Student Conference at the University of Illinois-Urbana Champaign, on "Lifting the Systems Ostrich's Head From the Sand: Introducing Ethical Systems."



May 2024

- Congratulations to PDL research partners Kaiyang Zhao and Hilbert (Yuang) Chen on receiving the Qualcaomm Innovation Fellowship.
- Wan Shen Lim interned with Microsoft Research's Data Systems Group in Redmond, WA this summer.
- Sarvesh Tandon interned at Oracle in the Data, Space, and Transactions organization, recently renamed to Mission-Critical Data and AI Engines.
- Hyoungjoo Kim interned at Microsoft Research this summer. His mentors were Qi Chen and Bailu Ding.
- Sam Arch interned with Amazon (NYC) working on Lambda UDFs for Amazon Redshift.
- Neharika Jali presented "Efficient Reinforcement Learning for Routing Jobs in Heterogeneous Queueing Systems" at the International Conference on Artificial Intelligence and Statistics (AISTATS), in Valencia, Spain.
- Nishant Ravi Shankar (INI) completed his Master's research on "SiaHet: Towards Exploiting Intra-Job Resource Heterogeneity in Heterogeneity-aware, Goodput Optimized Deep Learning Cluster Scheduling."

April 2024

- Student/advisor team Siyuan Chen and Phil Gibbons won their division in the CMU Random Distance Run.

continued on page 17

continued from page 1

A Call for Research on Storage Emissions

Sara McAllister, Fiodar Kazhamiaka, Daniel S. Berger, Rodrigo Fonseca, Kali Frost, Aaron Ogus, Maneesh Sah, Ricardo Bianchini, George Amvrosiadis, Nathan Beckmann, Gregory R. Ganger

HotCarbon'24, July 9, 2024, Santa Cruz, CA.

Major cloud providers have committed to lowering carbon emissions by 2030 across their datacenters, and research has contributed many ideas on how this may be achieved. However, a major contributor to datacenter emissions has not received enough attention: storage. Storage — everything from file storage to inter-application messaging in datacenters — causes 33% of operational emissions and 61% of embodied emissions in Azure's general-purpose cloud, based on a recent study. This paper identifies key sources of both operational and embodied emissions within distributed storage in datacenters. We also discuss strategies to reduce storage emissions and their challenges due to storage's fundamentally stateful nature.

The Holon Approach for Simultaneously Tuning Multiple Components in a Self-Driving Database Management System with Machine Learning via Synthesized Proto-Actions

William Zhang, Wan Shen Lim, Matthew Butrovich, Andrew Pavlo

Proceedings of the VLDB Endowment, 17(11): 3373-3387, 2024. July 2024.

Existing machine learning (ML) approaches to automatically optimize database management systems (DBMSs) only target a single configuration space at a time (e.g., knobs, query hints, indexes). Simultaneously tuning multiple configuration spaces is challenging due to the combined space's complexity. Previous tuning methods work around this by sequentially tuning individual spaces with a pool of tuners. However, these approaches struggle to coordinate their tuners and get stuck in local optima.

This paper presents the Proto-X framework that holistically tunes multiple configuration spaces. The key idea of Proto-X is to identify similarities across multiple spaces, encode them in a high-dimensional model, and then synthesize "proto-actions" to navigate the organized space for promising configurations. We evaluate Proto-X against state-of-the-art DBMS tuning frameworks on tuning PostgreSQL for analytical and transactional workloads. By reasoning about configuration spaces that are orders of magnitude more complex than other frameworks (both in terms of quantity and variety), Proto-X discovers configurations that improve PostgreSQL's performance by up to 53% over the next best approach.

Reducing Cross-Cloud/Region Costs with the Auto-Configuring MACARON Cache

Hojin Park, Ziyue Qiu, Gregory R. Ganger, George Amvrosiadis

SOSP '24, November 4–6, 2024, Austin, TX, USA.

An increasing demand for cross-cloud and cross-region data access is bringing forth challenges related to high data transfer costs and latency. In response, we introduce Macaron, an auto-configuring cache system designed to minimize cost for remote data access. A key insight behind Macaron is that cloud cache size is tied to cost, not hardware limits, shifting the way we think about cache design and eviction policies. Macaron dynamically configures cache size and utilizes a mix of cloud storage types, in order to adapt to workload changes and reduce cloud costs. We demonstrate that Macaron can reduce cross-cloud workload costs by 65% and cross-region costs by 67%, mainly by reducing outgoing data transfer and by leveraging object storage alongside DRAM to reduce cache capacity cost.

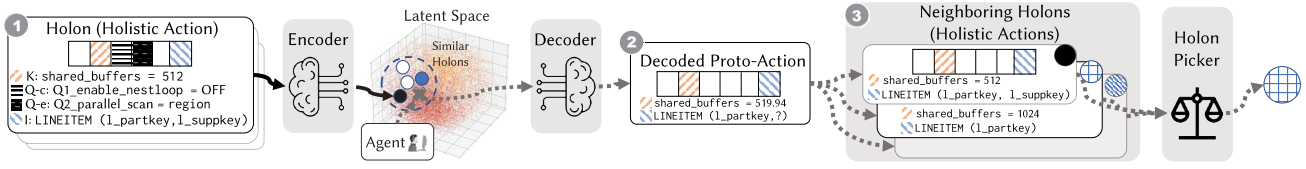
FairyWREN: A Sustainable Cache for Emerging Write-Read-Erase Flash Interfaces

Sara McAllister, Yucong "Sherry" Wang, Benjamin Berg*, Daniel S. Berger†, George Amvrosiadis, Nathan Beckmann, Gregory R. Ganger

18th USENIX Symposium on Operating Systems Design and Implementation (OSDI '24), July 10–12, 2024. Santa Clara, CA, USA.

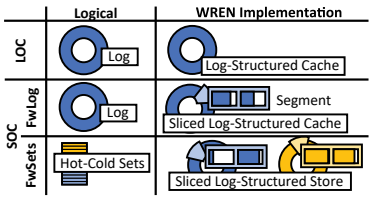
Datacenters need to reduce embodied carbon emissions, particularly for flash, which accounts for 40% of embodied carbon in servers. However, decreasing flash's embodied emissions is challenging due to flash's limited write endurance, which more than

continued on page 6



Proto-Actions and Neighborhoods – A holon passes through an encoder (e.g., neural network) to obtain a point (filled-in circle) in latent space where holons of similar performance or structure are nearby. A proto-action is a point (X) in latent space chosen by a tuning agent. Decoding the proto-action does not necessarily result in a valid holon. Instead, the agent searches the proto-action's neighborhood for a valid holon and selects the most promising one.

continued from page 5



The components of FairyWREN.

halves with each generation of denser flash. Reducing embodied emissions requires extending flash life-time, stressing its limited write endurance even further. The legacy Logical Block-Addressable Device (LBAD) inter- face exacerbates the problem by forcing devices to perform garbage collection, leading to even more writes.

Flash-based caches in particular write frequently, limiting the lifetimes and densities of the devices they use. These flash caches illustrate the need to break away from LBAD and switch to the new Write-Read-Erase iNterfaces (WREN) now coming to market. WREN affords applications control over data placement and garbage collection. We present FairyWREN1, a flash cache designed for WREN. FairyWREN reduces writes by co-designing caching policies and flash garbage collection. FairyWREN provides a 12.5× write reduction over state-of-the-art LBAD caches. This decrease in writes allows flash devices to last longer, decreasing flash cost by 35% and flash carbon emissions by 33%.

The Tyr Dataflow Architecture: Improving Locality by Taming Parallelism

Nikhil Agarwal, Mitchell Fream, Souradip Ghosh, Brian Schwedock, Nathan Beckmann. MICRO 2024

57th IEEE/ACM International Symposium on Microarchitecture, Nov 2–6, 2024, Austin, Texas.

Architectures should aim to maximize parallelism within a machine's finite memories, but prior designs tend to extremes, either maximizing parallelism or minimizing state. In particular, prior unordered dataflow architectures suffer from a parallelism explosion that creates unbounded state, requires prohibitively large associative memories, and risks deadlock. The few architectures that successfully navigate the parallelism-state tradeoff are limited to embarrassingly parallel programs.

TYR is a new, general-purpose unordered dataflow architecture that achieves high parallelism with bounded state. The key insight is that prior unordered dataflow architectures are overly conservative, unnecessarily allocating tags from a single, global tag space. TYR exploits program structure to break up tags into local tag spaces that operate independently. Local tag spaces eliminate tag competition between co-dependent parts of the program, provably guaranteeing forward progress with only two tags per local tag space. TYR thus opens the door to an efficient, scalable implementation of unordered dataflow. Simulation of parallel programs demonstrates that TYR achieves parallelism nearly identical to a naïve unordered dataflow architecture with orders-of-magnitude less state.

Erasure Coded Neural Network Inference via Fisher Averaging

Divyansh Jhunjhunwala, Neharika Jali, Gauri Joshi, Shiqiang Wang

IEEE International Symposium on Information Theory (ISIT), Athens, Greece, July 7-12, 2024.

Erasure-coded computing has been successfully used in cloud systems to reduce tail latency caused by factors such as straggling servers and heterogeneous traffic variations. A majority of cloud computing traffic now consists of inference on neural networks on shared resources where the response time of inference queries is also adversely affected by the same factors. However, current erasure coding techniques are largely focused on linear computations such as matrix-vector and matrix-matrix multiplications and hence do not work for the highly non-linear neural network functions. In this paper, we seek to design a method to code over neural networks, that is, given two or more neural network models, how to construct a coded model whose output is a linear combination of the outputs of the given neural networks. We formulate the problem as a KL barycenter problem and propose a practical algorithm COIN that leverages the diagonal Fisher information to create a coded model that approximately outputs the desired linear combination of outputs. We conduct experiments to perform erasure coding over neural networks trained on real-world vision datasets and show that the accuracy of the decoded outputs using COIN is significantly higher than other baselines while being extremely compute-efficient.

Morph: Efficient File-Lifetime Redundancy Management for Cluster File Systems

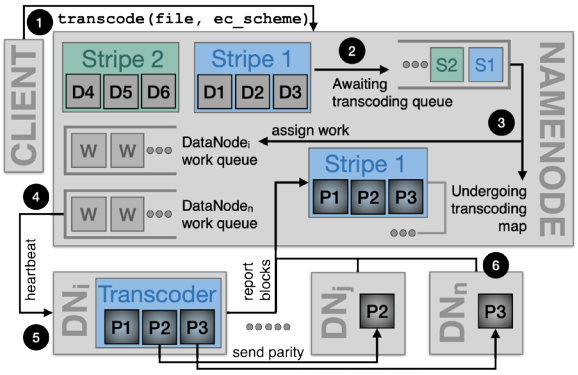
Timothy Kim, Sanjith Athlur, Saurabh Kadekodi, Francisco Maturana Dax Delvira, Arif Merchant, Gregory R. Ganger, K. V. Rashmi

SOSP '24, November 4–6, 2024, Austin, TX.

Many data services tune and change redundancy configurations of files over their lifetimes to address changes in data temperature and latency requirements. Unfortunately, changing redundancy configs (transcode) is IO-intensive. The Morph cluster file system introduces new transcode-efficient redundancy schemes to minimize overheads as files progress through lifetime phases. For newly ingested data, commonly stored via 3-way replication, Morph introduces a hybrid redundancy scheme that combines a replica with an erasurecoded (EC) stripe, reducing both ingest IO and capacity overheads while enabling free transcode to EC by deleting replicas. For subsequent transcodes to wider, more space-efficient EC configs,

continued on page 7

continued from page 7



Morph transcode architecture in the context of HDFS components. In this diagram, a file is transcoded into two new stripes, each with 3 data chunks and 3 new parity chunks. Note that the Namenode only handles the assigning of transcode work to Datanodes and does not actually read or write the file data.

Morph exploits Convertible Codes, which minimize data read for EC transcode, and introduces new block placement policies to maximize their effectiveness.

Analysis of data ingest and transcode activity in Google storage clusters shows the current massive IO load and the potential savings from Morph's approach—transcode IO can be reduced by over 95%, and total ingest+transcode IO can be reduced by 50–60% while also reducing capacity overheads for newly ingested data by 20%. Experiments evaluating a Morph implementation in HDFS show that these benefits can be realized in a real system without hidden increases in complexity, tail latency, or degraded-mode latency.

Data Caching for Enterprise-Grade Petabyte-Scale OLAP

Chunxu Tang, Bin Fan, Jing Zhao, Chen Liang, Yi Wang, Beinan Wang, Ziyue Qiu, Lu Qiu, Bowen Ding, Shouzhuo Sun, Saiguang Che, Jiaming Mai, Shouwei Chen, Yu Zhu, Jianjian Xie, Yutian (James) Sun, Yao Li, Yangjun Zhang, Ke Wang, Mingmin Chen

2024 USENIX Annual Technical Conference. July 10–12, 2024 • Santa Clara, CA, USA.

With the exponential growth of data and evolving use cases, petabyte-scale OLAP data platforms are increasingly adopting a model that decouples compute from storage. This shift, evident in organizations like Uber and Meta, introduces operational challenges including massive, read-heavy I/O traffic with potential throttling, as well as skewed and fragmented data access patterns. Addressing these challenges, this paper introduces the Alluxio local (edge) cache, a highly effective architectural optimization tailored for such environments. This embeddable cache, optimized for petabyte-scale data analytics, leverages local SSD resources to alleviate network I/O and API call pressures, significantly improving data transfer efficiency. Integrated with OLAP systems like Presto and storage services like HDFS, the Alluxio local cache has demonstrated its effectiveness in handling large-scale, enterprisegrade workloads over three years of deployment at Uber and Meta. We share insights and operational experiences in implementing these optimizations, providing valuable perspectives on managing modern, massive-scale OLAP workloads.

Designing Cloud Servers for Lower Carbon

Jaylen Wang, Daniel S. Berger, Fiodar Kazhamiaka, Celine Irvene, Chaojie Zhang, Esha Choukse, Kali Frost, Rodrigo Fonseca, Brijesh Warrier, Chetan Bansal, Jonathan Stern, Ricardo Bianchini, Akshitha Sriraman

Proceedings of the 51st Intl. Symposium on Computer Architecture (ISCA 2024), Buenos Aires, Argentina, June 2024.

To mitigate climate change, we must reduce carbon emissions from hyperscale cloud computing. We find that cloud compute servers cause the majority of emissions in a general-purpose cloud. Thus, we motivate designing carbon-efficient compute server SKUs, or GreenSKUs, using recently-available lowcarbon server components. To this end, we design and build three GreenSKUs using low-carbon components, such as energy-efficient CPUs, reused old DRAM via CXL, and reused old SSDs.

We detail several challenges that limit GreenSKUs' carbon savings at scale and may prevent their adoption by cloud providers. To address these challenges, we develop a novel methodology and associated framework, GSF (GreenSKU Framework), that enables a cloud provider to systematically evaluate a GreenSKU's carbon savings at scale. We implement GSF within Microsoft Azure's production constraints to evaluate our three GreenSKUs' carbon savings. Using GSF, we show that our most carbon-efficient GreenSKU reduces emissions per core by 28% compared to currently-deployed cloud servers. When designing GreenSKUs to meet applications' performance requirements, we reduce emissions by 15%. When incorporating overall data center overheads, our GreenSKU reduces Azure's net cloud emissions by 8%.

DéjàVu: KV-cache Streaming for Fast, Fault-tolerant Generative LLM Serving

Foteini Strati, Sara McAllister, Amar Phanishayee, Jakub Tarnawski, Ana Klimovic

41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, July 21-27, 2024.

Distributed LLM serving is costly and often underutilizes hardware accelerators due to three key challenges: bubbles in pipeline-parallel deploy-

continued on page 18

September 2024
Welcome Cecilia!

We are thrilled to announce the arrival of Greg's son's daughter Cecilia Ganger! Born in September to Tim and Julia, she is sleeping well, and bringing joy to everyone.

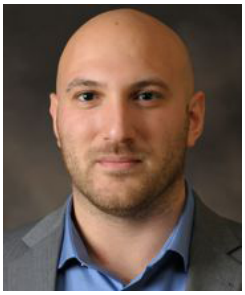
September 2024
Sara McAllister Named 2025 Siebel Scholar

Congratulations to Sara on becoming a Siebel Scholar! Her work on computer systems focuses on distributed, caching and storage systems, leveraging hardware-software co-design and grounding system design in mathematical modeling to enable more efficient and sustainable systems. McAllister also strongly supports diversity, equity and inclusion in computing. She co-created the CS-JEDI course (Intro to Justice, Equity, Diversity, and Inclusion in Computer Science) course that is now required for the computer science Ph.D. program.

The Siebel Scholar program was founded in 2000 by the Thomas and Stacey Siebel Foundation. It recognizes nearly 80 students each year whose work influences the technologies, policies, and economic and social decisions that shape the future.

-- info from CMU SCS News, Sept 20, 2024

September 2024
Dimitrios Skarlatos Recognized as Intel Rising Star



Congratulations to Dimitrios (Assistant Professor, SCS), who has been named an Intel Rising Star. His research bridges hardware and operating systems and delves into the core challenges of datacenter computing, addressing fundamental questions about scalability limitations, security vulnerabilities, and energy efficiency. His past work on memory management has tackled long-standing system design challenges at the interface of OS and hardware, which can severely impede server efficiency. His contributions at the algorithmic, OS, and hardware level have enabled highly efficient virtual memory and memory management for large-scale systems. These innovations have led to major gains in production data centers. Dimitiros' work further extends into the domain of security at the intersection of OS and hardware. He has uncovered vulnerabilities in the software-hardware interface and has designed comprehensive hardware and OS mechanisms to reduce the attack surface of operating systems.

The Intel Rising Star Faculty Award (RSA) program acknowledges eight early-career academic researchers leading ground-breaking technology research and facilitates collaboration between award winners and leaders at Intel. Those selected conduct research to find novel solutions to challenges spanning various topics, including artificial intelligence, computer architecture, quantum computing, manufacturing processing and packaging technology, security, and quantum photonics.

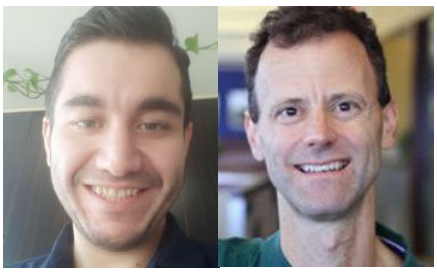
-- info from www.intel.com

July 2024
Akshitha Sriraman Receives George Tallman Ladd Research Award

Congratulations to Akshitha, Assistant Professor of Electrical and Computer Engineering, on receiving the CMU College of Engineering George Tallman Ladd Research Award! The award is made each year to one or more assistant professors in the College of Engineering who have not yet been considered for promotion to Associate Professor. Selection of the recipient is made in recognition of outstanding research and professional accomplishments and potential. The award, named for George T. Ladd, a trustee of the Carnegie Institute of Technology from 1938 until his death in 1943. The bequest of Ladd and his wife, Florence Barrett Ladd, formed a foundation that has funded faculty research in the College of Engineering at Carnegie Mellon for the past 70 years.

-- info from CMU College of Engineering

June 2024
PDL Paper Ties for Best Paper at SIGMETRICS '24!



Congratulations to PDL authors Mohammad Bakshalipour and advisor Phil Gibbons on their award for Best Paper at Sigmetrics 2024, held in Venice, Italy this summer. The paper "Agents of Autonomy: A Systematic Study of

Robotics on Modern Hardware" presents a systematic performance study of robotics on modern hardware and introduces RoWild, an open-source benchmark suite for robotics that is comprehensive and cross-platform, encompassing a broad range of robots, such as driverless vehicles, pilotless drones, and stationary robotic arms.

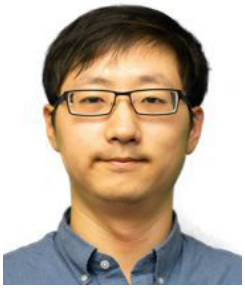
May 2024
PDL Students Win Qualcaomm Innovation Fellowship

Congratulations to PDL research partners Kaiyang Zhao (CS) and Hilbert (Yuang) Chen (ECE), who have been selected to receive a Qualcomm Innovation Fellowship for their work on Learned Virtual Memory for Heterogeneous Architectures. The project rethinks virtual memory using lightweight machine learning models to solve challenges in data centers and at the edge. Virtual memory has become a severe bottleneck due to the significant increase in memory-intensive workloads such as artificial intelligence applications. The ambitious goal of the project is to provide a pliable abstraction that dynamically learns and adjusts based on the workload behavior and underlying hardware. The result will enable major performance and energy efficiency gains, and ultimately reduce the carbon footprint across edge devices and servers in data centers.

Students selected for the Qualcomm Innovation Fellowship earn a one-year fellowship and are mentored by Qualcomm engineers to facilitate the success of the proposed research. The fellowship comes with $100,000 to fund that research.

-- from CMU SCS News, Aaron Aupperlee, Tuesday, May 14, 2024

April 2024
Community Award for Best Paper at NSDI!



Congratulations to Juncheng Yang, Rashmi Vinayak and their co-author on having their paper selected to receive the Community award for Best Paper at NSDI! The paper, "SIEVE is Simpler than LRU: An Efficient Turn-Key Eviction Algorithm for Web Caches" discusses new caching and eviction algorithms to maximize efficiency, reducing the cache miss ratio. This is the team's second community award - they also won in 2021 for their paper "Segcache: A Memory-efficient and Scalable In-memory Key-value Cache for Small Objects."

February 2024
Nathan Beckmann Receives 2024 Sloan Fellowship

Congratulations to Nathan Beckmann, who has been named a Sloan Research Fellow for 2024. Among 126 early-career scholars, Nathan represents the most promising scientific researchers working today. Their achievements and potential place them among the next generation of scientific leaders in the U.S. and Canada. Nathan's current research focuses on building energy-efficient, general-purpose, post-Von Neumann computers, showing that systems can be both general-purpose and highly energy-efficient.

January 2024
Akshitha Sriraman Wins NSF Career Award!

The National Science Foundation (NSF) has awarded Electrical and Computer Engineering Assistant Professor Akshitha Sriraman an NSF Faculty Early Career Development (CAREER) Award, a prestigious five-year grant given to junior faculty for research and education. Akshitha's research bridges computer architecture and software systems, with a focus on making hyperscale data center systems more efficient, sustainable, and equitable via solutions that span the systems stack. Her work has developed the software and hardware foundations of hyperscale data center systems that support modern web services, such as web search, video streaming, and online healthcare.

-- from CMU ECE News, January 17, 2024

November 2023
Sara McAllister Named EECS Rising Star!

PDL PhD student, Sara McAllister (Computer Science), advised by Nathan Beckmann and Greg Ganger was named an EECS Rising Star for her work exploring Efficient and Sustainable Data Retrieval Systems: Zettabytes of data, mostly stored in massive, hyperscale datacenters enable much of today's modern computing. However, datacenters are projected to account for 33% of the global carbon emissions by 2050. While most datacenters are moving to renewable energy, the majority of their emissions are not from energy generation, but rather from embodied emissions, generated from lifecycle events like manufacturing and mining raw materials. One key way to reduce embodied emissions is to increase device lifetime. Unfortunately, extending the lifetime of storage is challenging because storage hardware

continued on page 27

IN MEMORIAM

Erik Riedel – 52, of Scituate, MA, passed away on June 30, 2024, leaving behind his loving family: his wife Cheryl LG Riedel, his beloved children Emma, Jonah, Elena, and Ani, his parents Juergen and Gerda, and his brother Jens Riedel, as well as many friends among the PDL.

Erik was a builder and an innovator. His goals, through the companies and organizations he worked with (IT Renew, EMC, Dell, Seagate, HP), some of which he helped found (Works Together, Flax Computing), were to advise and assist companies and individuals through the challenges of developing complex, scalable distributed systems. He achieved successful results through effective and consistent everyday collaboration. Most recently, with Flax Computing in Boston, Erik's mission was decarbonization of data center computing - increasing efficiency, lowering costs, and reducing carbon footprints. Earlier in his career, Erik was Senior Director of Technology & Architecture in the Cloud Infrastructure Group at EMC in Cambridge, MA. He worked to build cloud storage technology for deployment in private and public clouds. Before joining EMC, Erik was Director of Interfaces & Architecture at Seagate Research in Pittsburgh, PA. The group he founded and led focused on novel storage devices and systems with increased intelligence to optimize performance, improve security and reliability, and enable smarter organization of data. The technology targeted both large-scale enterprise storage clusters and ad-hoc collections of consumer and mobile storage devices working together.

Erik held B.S., M.S.E. and Ph.D. degrees from Carnegie Mellon University. His thesis work was on Active Disks as an extension to Network-Attached Secure Disks (NASD). Among us, Erik was honored for his accomplishments in 2010 when he was named a PDL Distinguished Alumni.

Erik's accomplishments are many, and the list above is limited. More important than his education, or his career, or where he lived and when, is HOW he lived; the ineffable qualities that have connected all of us here now, in collecting our fondest memories and in sharing our sorrow. There are no words that can convey the shape and weight of the loss we are processing, but we are comforted in knowing that his compassion, his infectious laugh, and his willingness to jump in and help have impacted so many others. Just like us, we know that many of you will think of him when you look up and notice the clouds passing by; that you'll smile

remembering his infamous moves on any available dance floor; that you'll miss his seemingly-endless patience while providing free tech support; that you'll feel some of his joy at finding a really good empty box that's the perfect size to recycle something... somewhere... at some point in time. We know you'll perform small acts of kindness that often go unnoticed to honor his commitment to making this world a more tender and caring place. We know that you'll look at his children and see the pride and ferocious love he breathed into them and wonder at the strength of the bonds he built with each of them, bonds that will sustain us all while we learn to navigate our lives without him. We know that when things look dark and scary, you'll remember Erik's favorite quote, Look for the Helpers (https://www.youtube.com/watch?v=-LGHtc_D328). We know you'll reflect his eternal optimism, lend a hand, provide a soft shoulder, and step up to be the helper someone needs in his stead, to offer a glimpse of the hope he provided each of us, while he rests.

DISSERTATION ABSTRACT: Machine Learning for Flash Caching in Bulk Storage Systems

Daniel Lin-Kit Wong
Carnegie Mellon University, SCS

PhD Defense — August 23, 2024

Flash caches are used to reduce peak backend load for throughput-constrained data center services, reducing the total number of backend servers required. Bulk storage systems are a large-scale example, backed by high-capacity but low-throughput hard disks, and use flash caches to provide a cost-effective storage layer underlying everything from blobstores to data warehouses.

However, flash caches must manage their limited write endurance and limit the flash write rate to avoid premature wear-out. They do so via admission policies that filter cache insertions and maximize the workload-reduction value of each write.

I evaluate and demonstrate potential uses of ML in place of traditional heuristic cache management policies for flash caches in bulk storage systems. The most successful elements of my research are embodied in a flash cache system called Baleen, which uses coordinated ML admission and prefetching to reduce peak backend load. After learning painful lessons with early ML policy attempts, I exploit a new cache residency model (episodes) to guide model training. I focus on optimizing an end-to-end metric (Disk-head Time) that measures backend load more accurately than IO or byte miss rate. Evaluation using 7-day Meta traces from 7 storage clusters shows Baleen reducing Peak Disk-head Time (and backend hard disks required) by 12% over state-of-the-art policies for a fixed flash write rate.

I present a TCO (total cost of ownership) formula quantifying the costs of additional flash writes against reductions in Peak Disk-head Time in terms of flash drives and hard disks needed. Baleen-TCO chooses optimal flash write rates and reduces estimated TCO by 17%.

Workloads change over time, requiring that caches adapt to maintain performance. I present a strategy for peak load reduction that adapts selectivity to load levels. I evaluated workload drift and its impact on ML policy performance on 30-day Meta traces.

Baleen is the result of substantial exploration and experimentation with ML for caching. I present lessons learned from additional strategies considered and explain why they saw limited success on our workloads. These include improvements for ML eviction and more advanced ML models.

Code and traces are available via https://www.pdl.cmu.edu/CILES/.

DISSERTATION ABSTRACT: Designing Efficient and Scalable Cache Management Systems

Juncheng Yang
Carnegie Mellon University, SCS

PhD Defense — August 22, 2024

Software caches have been widely deployed at scale in today's computing infrastructure to improve data access latency and throughput. These caches consume PBs of DRAM at many companies, which necessitates high efficiency —achieving the same miss ratio with less DRAM consumption. Meanwhile, modern servers have hundreds of cores, making scalability a critical requirement for designing software caches. This thesis explores different approaches to improving the efficiency and scalability of software caches.

This thesis has two parts. The first part focuses on system designs that allow caches to store more objects in the cache to achieve a low miss ratio. In this part, I will describe three works. First, I will discuss what key-value cache workloads at Twitter look like using a large-scale workload analysis. Second, drawing on insights from the workload study, I will describe the design of Segcache, a TTL-indexed segment-structured key-value cache that quickly removes expired objects, provides tiny object metadata, and enables close-to-linear scalability. Third, I will present C2DN to demonstrate a fault-tolerant CDN cache cluster using erasure coding for low-overhead redundancy.

Brian Schwedock receives his PhD hood from Nathan Beckmann.

The second part focuses on algorithms that allow the cache to store more useful objects in the cache, which is also critical for cache efficiency. First, I will investigate the design of a low-overhead learned cache. Existing caches using machine learning often incur significant storage and computation overheads. I will show that learning on the group level amortizes overheads and accumulates more information for better learning. While GL-Cache is faster than existing learned caches, it is still more complex compared to simple heuristics. In the following chapter, I will discuss two techniques, lazy promotion and quick demotion, which enable us to design simple yet effective eviction algorithms. In the third chapter, I will discuss an example using the two techniques, S3-FIFO, a new eviction algorithm only composed of FIFO queues. In the last chapter, I will present SIEVE, a new eviction algorithm that uses one queue to achieve lazy promotion and quick demotion. SIEVE is simpler than LRU, but achieves state-of-the-art efficiency and scalability.

continued on page 12

continued from page 11

THESIS PROPOSAL: Efficient Job-resource Co-Adaptivity for Deep learning Workloads on Large Heterogeneous GPU Clusters

Suhas Jayaram Subramanya, CSD
August 6, 2024

The training performance of a deep learning (DL) training job is determined by the number, type and arrangement of the allocated GPU resources, and the job parameters (like batch size and learning rate) used for execution. Modern clusters for DL training contain tens of thousands of GPUs of many types, and a cluster scheduler allocates GPUs to training jobs to maximize collective training progress in the cluster. Existing DL cluster schedulers cannot handle the large space of adaptivity choices (i.e., combined space of GPU allocations and job parameters) for large, heterogeneous GPU clusters -- many are not heterogeneity-aware, few are adaptivity-aware, and none scale to large clusters without sacrificing allocation fidelity and cluster efficiency.

In this thesis, we introduce (a) a scheduler to facilitate efficient job-resource adaptivity for DL training jobs on large heterogeneous GPU clusters, and (b) a method to scale optimization-based scheduling to much larger cluster sizes without sacrificing allocation fidelity and resource efficiency. Our adaptivity-aware scheduler, Sia, uses GPU resources judiciously to learn a job's training performance across different GPU types, and continuously co-optimizes the GPU allocation and job execution parameters to maximize cluster-wide training progress in heterogeneous GPU clusters. We then scale Sia to large cluster sizes by modeling the scheduling policy as a continuous optimization problem. We show that it is possible to augment the interface between a scheduler and the optimization problem solver to efficiently track changes to the scheduling problem arising from changing cluster conditions like job arrivals, departures and phase changes. We develop a prototype solver with the augmented interface for the Sia scheduling policy that can efficiently recover allocations for very large clusters. As an additional contribution, we observe that many other resource-allocation problems can also be formulated as continuous optimization problems and can be solved both quickly and efficiently using our proposed solver.

DISSERTATION ABSTRACT: Verifying Concurrent Systems Code

Travis Hance
Carnegie Mellon University, SCS

PhD Defense — August 7, 2024

Concurrent software is notoriously difficult to write correctly, so to increase confidence in it, it is often desirable to apply formal verification techniques. One technique that is especially promising for verifying concurrent software is concurrent separation logic (CSL), which uses reasoning principles based on resource ownership. However, even with CSL, verifying complex systems at scale (e.g., those with 1000s of lines of code) remains challenging. The reasons it remains challenging include:

(1) The manual proof effort required by many existing CSL frameworks.

(2) The inherent complexity of the target systems. Sophisticated systems may have custom, low-level synchronization logic, which may be deeply intertwined with domain logic, in the interest of performance.

We posit that a promising way to overcome (1) is, rather than using CSL directly, to use an ownership type system such as Rust's, taking advantage of its sophisticated but efficient type-checking algorithms. To demonstrate this, we develop a full methodology, from theory to implementation, based around this core idea, showing that we can recover the rich reasoning principles of CSL in this setting. In particular, we show that this methodology is rich enough to support the verification of inherently complex systems as in (2).

Part of Greg Ganger's Gang of Grad Students. From L to R: Daniel Wong, Sara McAllister, Hojin Park, GG, Daiyaan Arfeen, Suhas J. Subramanya, and Sanjith Athlur. Missing are Ziyue Qiu and Timothy Kim.

MASTERS THESIS: SiaHet: Towards Exploiting Intra-Job Resource Heterogeneity in Heterogeneity-aware, Goodput Optimized Deep Learning Cluster Scheduling

Nishant Ravi Shankar, INI
May 31, 2024

The Sia scheduler represents an advancement in efficiently allocating cluster resources to Deep Learning Training jobs in a heterogeneous GPU cluster,

continued on page 13

continued from page 12

resulting in improved Job Completion Times (JCTs) and cluster utilization. However, its current implementation only addresses GPU heterogeneity in its allocation decisions, while eventually choosing to assign only homogeneous GPU resources for a Deep Learning Training job. This limitation highlights the significance of exploring Intra-Job Heterogeneity during scheduling decisions, which can unlock more cluster utilization and parallelism, thereby further optimizing average JCTs. The aim of the thesis is to address Sia scheduler's limitation by developing SiaHet, an enhancement over the Sia Scheduler, that proposes two key features to unlock Intra-Job Resource Heterogeneity: a) An enhanced heterogeneous resource allocation policy for Deep Learning Training jobs b) an execution engine runtime that is capable of performing hybrid parallel Deep Learning execution over heterogeneous resources. This thesis also aims to study the effects of Intra-Job Resource Heterogeneity through workload experiments in a small-scale research cluster, demonstrating their benefits on mixed priority workloads on specific cluster sizes.

MASTERS THESIS: Towards an OS for GPUs: Threadblock Scheduling for Deep Learning Workloads

Brian Zhang, CSD
April 24, 2024

As the year over year performance gains of CPUs has stagnated with the death of Moore's Law, GPUs and other data parallel chips have seen a surge in demand particularly for use in datacenter deep learning workloads. In spite of the growing demand, many companies are unable to fully utilize the hardware that is already in their datacenters. In fact, Alibaba reported a median GPU utilization of less than 10% in 2020. This number implies vast over-provisioning and shows the benefits to be gained via GPU multi-tenancy.

Just as multi-tenancy with traditional CPU architectures is facilitated with an OS, we believe that an OS can similarly solve this problem for GPUs. In this thesis we describe the design and implementation of the compute scheduler of AxOS, an OS for data parallel accelerators. AxOS allows for transparency, high GPU utilization, performance isolation, and spatial stacking between multiple processes using the GPU. To achieve this, AxOS has a novel threadblock-centric approach to GPU compute scheduling via the virtual streams abstraction, kernel chunking, and rightsizing. We evaluate AxOS on a number of deep learning workloads to show these benefits.

DISSERTATION ABSTRACT: Building Reliable and Transparent Machine Learning Systems Using Structured Intermediate Representations

Giulio Zhe Zhou
Carnegie Mellon University, SCS

PhD Defense — April 3, 2024

Machine learning (ML) is increasingly used to drive complex applications such as web-scale search, content recommendation, autonomous vehicles, and language-based digital assistants. In recent years, these systems have become predominantly data-driven, often underpinned by deep learning models that learn complex functions end-to-end from large amounts of available data. But their purely data-driven nature also makes the learned solutions opaque, sample inefficient, and brittle.

To improve reliability, production solutions often take the form of ML systems that leverage the strengths of deep learning models while handling auxiliary functions such as planning, validation, decision logic, and policy compliance using other components of the system. However, because these methods are often applied post-hoc on fully trained, blackbox deep learning models, their ability to improve system reliability and transparency is limited.

Several 2023 PDL Retreat attendees gather for a round of Mario Kart after an evening poster session.

In this thesis, we study how to build more reliable and transparent ML systems using ML models with structured intermediate representations (StructIRs). Compared to non-structured representations such as neural network activations, StructIRs are directly obtained by optimizing a well-defined objective and are structurally constrained (e.g., to normalized embeddings or compilable code) while remaining sufficiently expressive for downstream tasks. They can thus make the resulting ML system more reliable and transparent by increasing modularity and making modeling assumptions explicit.

We explore the role of StructIRs in three different ML systems. In our first work, we use simple probability distributions parameterized by neural networks to build an effective ML-driven datacenter storage policy. In our second work, we show that grounding text generation in a well-structured vector embedding space enables effective transformation of high-level text attributes such as tense and sentiment with simple, interpretable vector arithmetic. In our final work, we conduct human subject studies showing that the stationarity assumptions behind bandit-based recommender systems do not hold in practice, demonstrating the importance of validating the assumptions and structures underlying ML systems.

continued on page 14

continued from page 13

DISSERTATION ABSTRACT: On Embedding Database Management System Logic in Operating Systems via Restricted Programming Environments

Matt Butrovich
Carnegie Mellon University, SCS

PhD Defense — April 5, 2024

The rise in computer storage and network performance means that disk I/O and network communication are often no longer bottlenecks in database management systems (DBMSs). Instead, the overheads associated with operating system (OS) services (e.g., system calls, thread scheduling, and data movement from kernel-space) limit query processing responsiveness. User-space applications can elide these overheads with a kernel-bypass design. However, extracting benefits from kernel-bypass frameworks is challenging, and the libraries are incompatible with standard deployment and debugging tools.

This thesis presents an alternative in user-bypass: a design that extends OS behavior for DBMS-specific features, including observability, networking, and query execution. Historically, DBMS developers avoid kernel extensions for safety and security reasons, but recent improvements in OS extensibility present new opportunities. With user-bypass, developers write safe, event-driven programs to push DBMS logic into the kernel and avoid user-space overheads. There are two ways to invoke user-bypass logic: (1) when a DBMS in user-space invokes these programs, user-bypass provides behavior similar to a new OS system call, albeit without kernel modifications. In contrast, (2) when an OS thread or interrupt triggers these programs in kernel-space, user-bypass inserts DBMS logic into the kernel stack.

First, we present a framework that employs user-bypass to collect training data for self-driving DBMSs efficiently. User-bypass programs reduce the number of round trips to kernel-space to retrieve performance counters and other system metrics. Next, we present a database proxy that applies user-bypass to support features like connection pooling and workload replication while reducing data copying and user-space thread scheduling. User-bypass programs embed DBMS network protocol logic in multiple layers of the OS network stack, applying DBMS proxy logic in a kernel-space fast path. Lastly, we present an embedded DBMS for future user-bypass applications. We discuss the design decisions, environment challenges, and performance characteristics of a DBMS that offers ACID transactions over multi-versioned data in kernel-space. We also explore applications of this user-bypass DBMS and compare them to modern user-space systems.

The techniques proposed in this thesis show user-bypass benefits across multiple DBMS design disciplines and provide a template for future DBMS and OS co-design.

Jason Boles operating the AV console at the 2023 PDL Retreat.

THESIS PROPOSAL: Securing Middleboxes Against Temporal Algorithmic Complexity Attacks

Nirav Atre, CSD
March 15, 2024

Denial-of-Service (DoS) attacks are the bane of public-facing network deployments. Temporal algorithmic complexity attacks (t-ACAs) are a class of DoS attacks where an attacker uses a small amount of adversarial traffic to induce a large amount of work in the target system, pushing the system into overload and causing it to drop packets from innocent users. t-ACAs are particularly dangerous because, unlike volumetric DoS attacks, they don't require a significant network bandwidth investment from the attacker. Today, middleboxes on the Internet must be designed and engineered on a case-by-case basis to mitigate the debilitating impact of t-ACAs; worse, the resulting designs tend to be overly conservative in their attack mitigation strategy, either throttling the middlebox's common-case performance, limiting the innocent traffic that it can serve, or both.

In this work, we propose the first general, systematic approach to make middleboxes resilient to t-ACAs. We design a framework, called SurgeProtector, that uses packet scheduling to mitigate the impact of t-ACAs using a well-known scheduling algorithm: Weighted Shortest Job First (WSJF). To evaluate SurgeProtector, we propose a new metric of vulnerability called the Displacement Factor (DF), which quantifies the "harm per unit effort" that an adversary can induce. We provide novel, adversarial analysis of WSJF and show that any system using this policy has a worst-case DF of only a small constant, where traditional schedulers place no upper bound on the DF.

Applying job size-based scheduling in an adversarial context also requires us to harden against attacks: (1) the scheduler's priority queue, and (2) heuristics used for job size estimation. To that end, we present a novel, adversary-proof hardware priority queue architecture, BBQ, that achieves 3X the packet processing rate of state-of-the-art hardware priority queue designs. Finally, we propose Cassandra, a tool to automatically generate adversary-resistant heuristics for arbitrary middleboxes. Illustrating that our framework is not only theoretically, but practically robust, we integrate SurgeProtector (with BBQ) into an open source intrusion detection system

continued on page 15

continued from page 14

(IDS). Under simulated attack, the SurgeProtector-augmented IDS suffers 90-99% lower innocent traffic loss than the original system.

THESIS PROPOSAL: Efficient and Sustainable Data Retrieval at Scale

Sara McAllister, CSD
February 19, 2024

Datacenters are projected to account for 33% of the global carbon emissions by 2050. As datacenters increasingly rely on renewable energy for power, the majority of datacenter emissions will be embodied — emissions from lifecycle stages including acquiring raw materials, manufacturing, transportation, and disposal. To reach the ambitious emission reduction goals set by both companies and governments, datacenters need to reduce emissions throughout their operations, including (and particularly relevant for this thesis) the storage system. Unfortunately, while data storage and retrieval systems are large contributors to embodied emissions, reducing their embodied emissions have largely been overlooked.

This thesis aims to address reducing emissions in data retrieval for large-scale storage systems. These systems can reduce their carbon footprint by enabling storage devices to have longer lifetimes and use denser media. However, storage hardware's IO limits combined with software's unnecessary additional IO often severely restrict emission reductions, or at worst cause increased emissions. Thus, this thesis focuses on reducing IO in several parts of the storage stack to enable efficient and sustainable data retrieval.

Sara McAllister offers her talk on "FairyWREN: A Sustainable Cache for Write-Read-Erase Interfaces" at last year's PDL Retreat.

First, this proposal addresses the efficiency and sustainability of flash caching, a critical layer in datacenter storage systems that is limited by flash write endurance. This improvement will be realized in two caching systems: Kangaroo and FairyWREN. Together, these caches dramatically reduce writes by over 28x, allowing flash devices to use denser flash for longer lifetimes, ultimately reducing emissions. Then, this thesis will discuss proposed work to enable more sustainable bulk storage, where bandwidth limitations prevent deployment of denser HDDs. I propose a new IO interface, Declarative IO, that empowers the storage system to eliminate duplicate IO accesses through exposing the time- and order-flexibility in maintenance tasks. This work will enable using larger HDDs, further reducing emissions from storage systems.

THESIS PROPOSAL: Receiver-assisted Congestion Control

Christopher Canel, CSD
December 12, 2023

Current techniques for rate control in computer networks are not aligned with the policies and considerations of both endpoints in a connection. Historically, fine-grained rate control has been the sender's responsibility through the Transmission Control Protocol (TCP) and its congestion control algorithm (CCA). However, the receiver, either due to conflicting priorities or greater visibility into congestion, would often be better served by a different rate allocation than the sender. A simple example is a many-flow incast in a datacenter network: each sender independently seeks to transmit as quickly as possible, while the receiver can observe that each flow should converge to a small fraction of the last-hop link rate. Likewise, on the Internet, each service attempts to maximize its own throughput, whereas a user may desire fairness across services.

To bridge the gap between endpoint objectives, we argue for a cooperative approach to congestion control that incorporates the receiver into the rate decision as well. Specifically, we advocate for receiver-assisted congestion control, where the receiver provides lightweight hints to senders about the rate regime in which they should operate. Receiver-assisted congestion control differs from fully receiver-based techniques because the sender maintains control over the packet stream, and our proposal offers capabilities similar to in-network rate control with fewer practical challenges. To implement receiver assistance, we revisit the well-known technique of TCP flow control and show it to be a powerful primitive that does not require modifications to TCP, the sender's CCA, or applications. This thesis proposal explores three case studies with differing endpoint objectives to show that receiver-assisted congestion control can improve Internet fairness, reduce packet loss in datacenters, and enable thousand-flow incast while reducing CPU overheads.

MASTERS THESIS: Survey and Evaluation of Database Management System Extensibility

Abigale Kim, CSD
December 13, 2023

Database management system (DBMS) extensibility is a feature which enables users to extend the DBMS with user software. However, the DBMS extensibility environment is fraught with perils, and DBMS developers have to resort to unspecified methods of developing extensions, including copying core DBMS source code and casing between different versions of the DBMS. Extending a DBMS to support new functionality is challenging due to the tight coupling between

continued on page 16

Kevin Gaffney, advised by Jignesh Patel, presents his PhD research on "AceHash: Performance-Focused Perfect Hashing for Database Systems" at the 2023 PDL Retreat.

continued from page 15

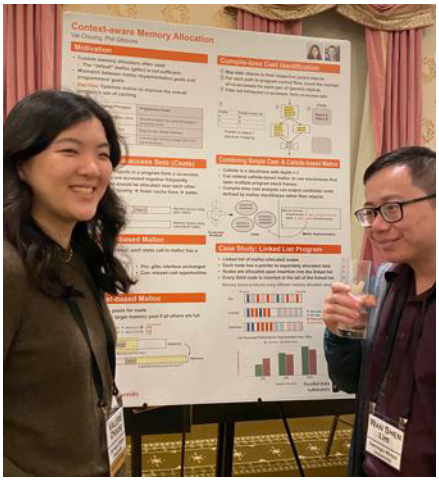
the system's internal components. This thesis studies and evaluates the design of DBMS extensibility. We first provide a comprehensive taxonomy of the types of extensibility supported by DBMSs and the effects of supporting their functionality within the DBMS. Given that PostgreSQL has the most variegated extensibility ecosystem, we also provide an in-depth analysis of it, where we evaluate how composable extensions were with one another, extension source code quality, and extension complexity. To assist us with this evaluation, we introduce an automated PostgreSQL extension analysis framework that collects information on how an extension integrates into the DBMS. We present results from static and dynamic analysis for over 100 extensions. We show correlations between the lack of composability of extensions and several factors related to their complexity and source code. We conclude with a discussion of the design decisions and trade-offs with supporting extensions in a DBMS.

THESIS PROPOSAL: Mitigating Fragility in ML Systems using Structured Models

Giulio Zhou, CSD
September 18, 2023

Machine learning (ML) is increasingly used to drive applications in a variety of complex settings, such as web-scale search, content recommendation, autonomous vehicles, and language-based digital assistants. These ML systems have, in recent years, become predominantly data-driven, often underpinned by deep learning models that can effectively learn complex functions end-to-end from large amounts of available data. Since they make few assumptions on the learned function, neural networks are a flexible and effective tool for a wide range of tasks and environments. However, their purely data-driven nature also makes the learned solutions opaque, sample inefficient, and potentially brittle. To address these problems, there has been much work on imposing structure into ML models. Some approaches impose structure implicitly (e.g. through architecture design and data augmentation), while others impose structure explicitly, such as by incorporating latent priors, geometric constraints, and physical models. When suitably applied, imposing explicit structure takes advantage of the powerful learning capabilities of deep learning models while avoiding the shortcomings of their end-to-end and blackbox nature. Imposing such structure into models also improves their transparency and yields a framework to represent various characteristics integral to the modeled task or environment, such as variability, periodicity, stationarity, smoothness, and monotonicity.

In this thesis, we explore approaches to improving the reliability and transparency of ML systems by imposing explicit structure in the form of simple parametric models. Compared to previous approaches that incorporate known domain-specific structure (e.g. based on physical models), we show that simple



Valerie Choung and Wan Shen Lim discuss Val's research on "Context-Aware Memory Allocation" during a PDL Retreat poster session.

parametric models are widely applicable, i.e. to any system whose behavior can be well-approximated by these models. We explore this using three case studies in different ML systems. In our first work, we show how to build an effective ML-driven storage system by modeling the variability of large warehouse-scale storage systems using univariate log-normal distributions (parametrized by neural networks.) In our second work, we conduct user studies to show that the typical assumption of stationary rewards in bandit-based recommender systems does not hold in practice, demonstrating the importance of validating the structures underlying ML systems. Lastly, for our proposed work, we analyze the benefits of imposing latent space structure in variational auto-encoders (VAEs) models of text. To improve the reliability of semantic transformations (such as changing tense and sentiment), we propose quantifying the isotropy and smoothness of the VAE latent space and exploring transformation methods that take advantage of its unique geometry.

continued from page 4

- Juncheng Yang, Rashmi Vinayak and their co-authors had their paper selected to receive the Community award for Best Paper for their work on "SIEVE is Simpler than LRU: An Efficient Turn-Key Eviction Algorithm for Web Caches" at NSDI!
- Nirav Atre presented "BBQ: A Fast and Scalable Integer Priority Queue for Hardware Packet Scheduling" at NSDI' 24 in Santa Clara, CA.
- Brian Zhang presented his MS research: "Towards an OS for GPUs: Threadblock Scheduling for Deep Learning Workloads."
- Giulio Zhe Zhou defended his PhD dissertation on "Building Reliable and Transparent Machine Learning Systems Using Structured Intermediate Representations."
- Matt Butrovich defended his PhD research: "On Embedding Database Management System Logic in Operating Systems via Restricted Programming Environments."
- Mingkuan Xu completed his speaking skill requirement, presenting "Quartz: Superoptimization of Quantum Circuits."
- Suhas J. Subramanya gave his speaking skills talk: "Sia: Heterogeneity-aware, Goodput-optimized ML-cluster Scheduling."

March 2024

- Hojin Park completed his speaking skills requirement, presenting "MACARON: Multi-cloud/region Aware Cache Auto-ReconfiguratiON."

Abigale Kim prepares her 2023 PDL Retreat talk on "Database Management System Extensibility."

- Giulio Zhou gave his speaking skills talk on "Learning on Distributed Traces for Datacenter Storage Systems."
- Nirav Atre proposed his PhD research on "Securing Middleboxes Against Temporal Algorithmic Complexity Attacks."

February 2024

- Nathan Beckmann received a 2024 Sloan Fellowship.
- Daniel Lin-Kit Wong presented "Baleen: ML Admission & Prefetching for Flash Caches" at FAST'24 in Santa Clara, CA.
- Sara McAllister proposed her PhD thesis topic "Efficient and Sustainable Data Retrieval at Scale."

January 2024

- Daniel Wong offered his speaking skills talk: "Baleen: ML Admission & Prefetching for Flash Caches."
- Akshitha Sriraman won an NSF Career Award!
- Sam Arch spoke at CIDR 2024, presenting "Dear User-Defined Functions, Inlining isn't working out so great for us. Let's try batching to make our relationship work. Sincerely, SQL"

December 2023

- Christopher Canel proposed his PhD research, titled "Receiver-assisted Congestion Control."
- Abigale Kim presented her Master's research: "Survey and Evaluation of Database Management System Extensibility."

November 2023

- The 29th PDL Retreat
- Sara McAllister named an EECS Rising Star!
- Niraj Tolia received the 2023 Parallel Data Lab Distinguished Alumni Award!
- Congratulations to CSD faculty members Todd Mowry and Nathan Beckmann on being named to the MICRO Hall of Fame!

October 2023

- Ziqi Wang presented "Memento: Architectural Support for Ephemeral Memory Management in Serverless Environments" at the 56th Annual IEEE/ACM International Symposium on Microarchitecture in Toronto, Canada.
- Juncheng Yang spoke on "FIFO Queues Are All You Need for Cache Eviction" at SOSP '23 in Koblenz, Germany. At the same conference, Jayaram Subramanya presented "Sia: Heterogeneity-aware, Goodput-optimized ML-cluster Scheduling."

PDL ALUMNI NEWS

John Linwood Griffin & Jiri Schindler

both PDL 1998-2004

Both John and Jiri and families still enjoy a yearly ski trip with retired PDC member Paul Massiglia, who was with Veritas when he joined the PDL at retreats and visit days for many years.

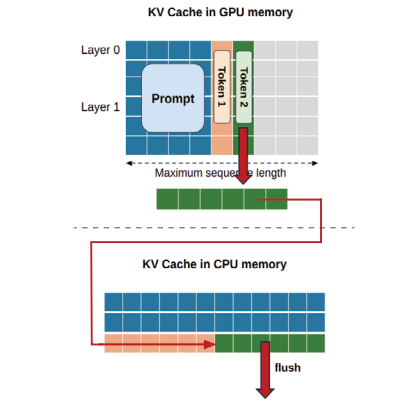
Andy Klosterman

PDL 1998 - 2009

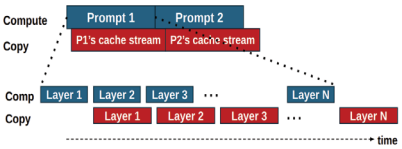
Andy recently discoverd that it was possible for him to claim dual Italian citizenship through his Grandmother. While he was between jobs, he worked on relocating his family to Bitritto, Italy. He and his family found a place to buy in his grandparents' hometown. They arrived at the end of July and their citizenship was recognized in early November.

continued from page 7

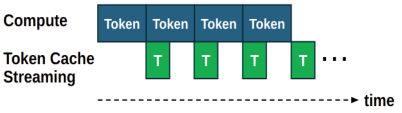
ments caused by the bimodal latency of prompt and token processing, GPU memory overprovisioning, and long recovery times in case of failures. DéjàVu addresses all these challenges using a versatile and efficient KV cache streaming library (DéjàVuLib). Using DéjàVuLib, we propose and implement efficient prompt-token disaggregation to reduce pipeline bubbles, microbatch swapping for efficient GPU memory management, and state replication for fault-tolerance. We highlight the efficacy of these solutions on a range of large models across cloud deployments.



(a) Buffered Copies: We aggregate small updates in a contiguous buffer in the GPU and then copy this buffer out.



(b) Layer-by-layer pipelining of prompt KV cache streaming with computation. We also pipeline the streaming of microbatch i with prompt processing of microbatch i + 1.



(c) Pipelining of token streaming with computation.

DejaVuLib KV cache streaming optimizations.

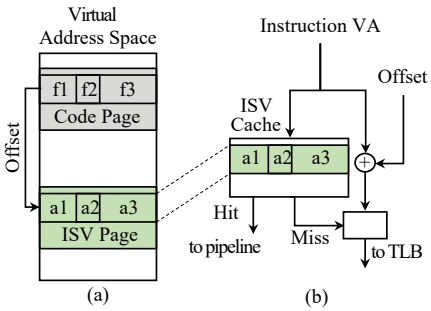
Perspective: A Principled Framework for Pliable and Secure Speculation in Operating Systems

Tae Hoon Kim, David Rudo, Kaiyang Zhao, Zirui Neil Zhao, Dimitrios Skarlatos

ISCA 2024, Buenos Aires, Argentina, June 2024.

Transient execution attacks present an unprecedented threat to computing systems. Protecting the operating system (OS) is exceptionally challenging because a transient execution gadget in the OS can potentially leak the entire memory.

In this work, we propose Perspective, a principled framework for building pliable and secure speculative execution defenses for the OS. Perspective offers a pliable interface that allows the OS to communicate its security requirements to hardware defenses, enabling tailored protection against transient execution attacks with little performance overhead. The design of Perspective is driven by a taxonomy of transient execution attacks in the OS kernel: (i) active transient execution attacks, where the attacker process exploits its own kernel thread to speculatively execute a transient execution gadget in the kernel, and (ii) passive transient execution attacks, where the attacker coerces the victim process's kernel thread to execute a transient execution gadget. Based on the taxonomy, Perspective introduces Data Speculation Views (DSVs) and Instruction Speculation Views (ISVs), to mitigate active and passive attacks, respectively. DSVs define the ownership of kernel data by a given execution context and block any speculative access to data outside the DSV. ISVs define the set of kernel functions that can be speculatively executed by a given execution context. Any transmitter instructions—whose execution could leak secrets, such as load instructions—that belong to kernel functions outside



Perspective's (a) ISV VA layout and (b) ISV hardware cache.

the ISVs are blocked from speculative execution. ISVs open up new opportunities of (i) swiftly patching gadgets in the OS, (ii) reducing the surface of passive attacks, and (iii) speeding up the process of auditing transient execution gadgets in the OS.

We build Perspective's software components in the Linux kernel and model the hardware components in gem5. We evaluate the security and performance of Perspective on a set of microbenchmarks and datacenter applications. Perspective has an execution overhead over an unprotected kernel of only 3.5% on microbenchmarks and only 1.2% on datacenter applications.

Helix: Distributed Serving of Large Language Models via Max-Flow on Heterogeneous GPUs

Yixuan Mei, Yonghao Zhuang, Xupeng Miao, Juncheng Yang, Zhihao Jia, Rashmi Vinayak

arXiv:2406.01566v1 [cs.DC] 3 Jun 2024.

This paper introduces Helix, a distributed system for high-throughput, low-latency large language model (LLM) serving on heterogeneous GPU clusters. A key idea behind Helix is to formulate inference computation of LLMs over heterogeneous GPUs and network connections as a max-

continued on page 19

continued from page 18

flow problem for a directed, weighted graph, whose nodes represent GPU instances and edges capture both GPU and network heterogeneity through their capacities. Helix then uses a mixed integer linear programming (MILP) algorithm to discover highly optimized strategies to serve LLMs. This approach allows Helix to jointly optimize model placement and request scheduling, two highly entangled tasks in heterogeneous LLM serving. Our evaluation on several heterogeneous cluster settings ranging from 24 to 42 GPU nodes show that Helix improves serving throughput by up to 2.7× and reduces prompting and decoding latency by up to 2.8× and 1.3×, respectively, compared to best existing approaches.

Agents of Autonomy: A Systematic Study of Robotics on Modern Hardware

Mohammad Bakhshalipour, Phillip B. Gibbons

SIGMETRICS/PERFORMANCE Abstracts '24, June 10–14, 2024, Venice, Italy. ACM, New York, NY, USA.

BEST PAPER AWARD!

As robots increasingly permeate modern society, it is crucial for the system and hardware research community to bridge its long-standing gap with robotics. This divide has persisted due to the lack of (i) a systematic performance evaluation of robotics on different computing platforms and (ii) a comprehensive, open-source, cross-platform benchmark suite.

To address these gaps, we present a systematic performance study of robotics on modern hardware and introduce RoWild, an open-source benchmark suite for robotics that is comprehensive and cross-platform. Our workloads encompass a broad range of robots, including driverless vehicles, pilotless drones, and stationary robotic arms, and we evaluate their performance on a spectrum of modern computing platforms, from low-end embedded CPUs to high-end server-grade GPUs.

Our findings reveal that current architectures experience significant inefficiencies when executing robotic workloads, highlighting the need for architectural advancements. We discuss approaches for meeting these requirements, offering insights for improving the performance of robotics. The full version of the paper is available in [11], and the source code of the benchmark suite is available in [2].

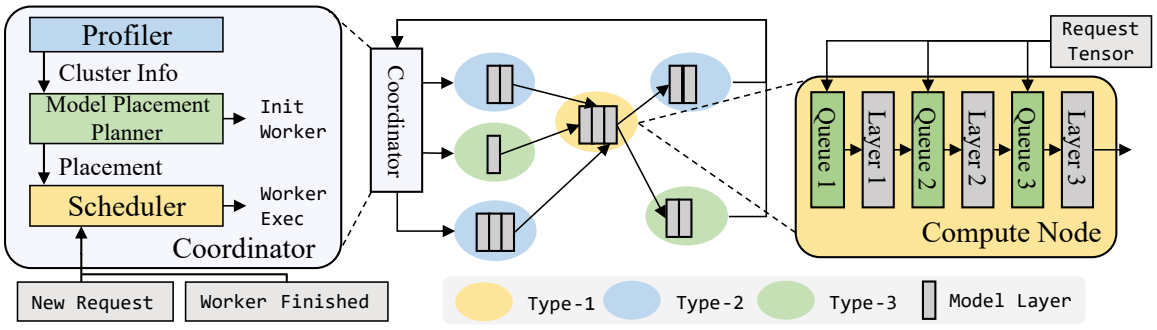
Efficient Reinforcement Learning for Routing Jobs in Heterogeneous Queueing Systems

Neharika Jali, Guannan Qu, Weina Wang, Gauri Joshi

Int'l Conference on Artificial Intelligence and Statistics (AISTATS), May 2nd - May 4th, 2024, Valencia, Spain.

We consider the problem of efficiently routing jobs that arrive into a central queue to a system of heterogeneous servers. Unlike homogeneous systems, a threshold policy, that routes jobs to the slow server(s) when the queue length exceeds a certain threshold, is known to be optimal for the one-fast-one-slow two-server system. But an optimal policy for the multi-server system is unknown and nontrivial to find. While Reinforcement Learning (RL) has been recognized to have great potential for learning policies in such cases, our problem has an exponentially large state space size, rendering standard RL inefficient. In this work, we propose ACHQ, an efficient policy gradient based algorithm with a low dimensional soft threshold policy parameterization that leverages the underlying queueing structure. We provide stationary-point convergence guarantees for the general case and despite the low-dimensional parameterization prove that ACHQ converges to an approximate global optimum for the special case of two servers. Simulations demonstrate an improvement in expected response time of up to ~30% over the greedy policy that routes to the fastest available server.

cont. on page 20



Helix overview. In Helix, the coordinator plans model placement as described in Sec. 3.3. We only need to run model placement once for each cluster. When a new request arrives, the coordinator node runs Helix scheduler to assign it a per-request pipeline and sends the request to the first node in the pipeline. Each compute node in the pipeline performs inference on the request on the layers it is responsible for and sends the (output for the) request to the next node in the pipeline. When the last node in the pipeline finishes performing inference on its layers, it will send the output token for the request to the coordinator (Worker Finished). The coordinator schedules generation of the next token for the request using the same pipeline.

continued from page 19

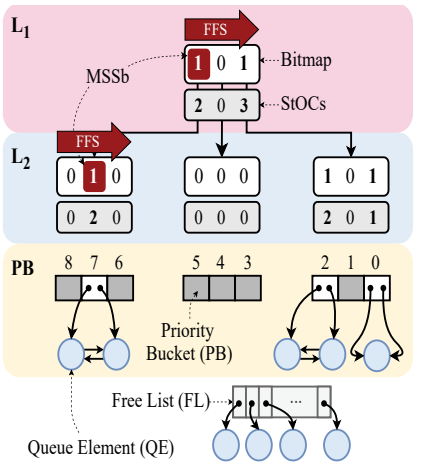
BBQ: A Fast and Scalable Integer Priority Queue for Hardware Packet Scheduling

Nirav Atre, Hugo Sadok, Justine Sherry

21st USENIX Symposium on Networked Systems Design and Implementation (NSDI' 24), April 16–18, 2024. Santa Clara, CA.

The need for fairness, strong isolation, and fine-grained control over network traffic in multi-tenant cloud settings has engendered a rich literature on packet scheduling in switches and programmable hardware. Recent proposals for hardware scheduling primitives (PIFO, PIEO, BMW-Tree, etc.) have enabled run-time programmable packet schedulers, considerably expanding the suite of scheduling policies that can be applied to network traffic. However, no existing solution can be practically deployed on modern switches and NICs because they either do not scale to the number of elements required by these devices or fail to deliver good throughput, thus requiring an impractical number of replicas.

In this work, we ask: is it possible to achieve priority packet scheduling at line-rate while supporting a large number of flows? Our key insight is to leverage a scheduling primitive used previously in software — called Hierarchical Find First Set — and port this to a highly pipeline-parallel hardware design. We present the architecture and implementation of Bitmapped Bucket Queue (BBQ), a hardware-based integer priority queue that supports a wide range of scheduling policies (via a PIFO-like abstraction). BBQ, for the first time, supports hundreds of thousands of concurrent flows while guaranteeing 100Gbps line rate (148.8 Mpps) on FPGAs and 1Tbps (1,488 Mpps) line rate on ASICs. We demonstrate this by implementing BBQ on a commodity FPGA where it is capable of supporting 100K+ flows and 32K+ priorities at 300MHz, 3X the packet rate of similar hardware priority queue designs. On ASIC, we can synthesize 100k elements at 3.1 GHz using a 7nm process.



2-level BBQ with w = 3 bit bitmaps. To dequeue the highest-priority element, we recursively perform FFS at each level of the tree starting with the root, following the most-significant set bit (MSSb) until we arrive at the required priority bucket.

Is Perfect Hashing Practical for OLAP Systems?

Kevin P. Gaffney, Jignesh M. Patel

4th Annual Conf. on Innovative Data Systems Research (CIDR '24) January 14-17, 2024, Chaminade, USA.

A perfect hash function (PHF) maps a set of keys to a range of integers with no collisions. Compared to conventional hash methods, PHFs are attractive for their low space overhead and reduced control flow. Despite their advantages, there has been little investigation into the use of PHFs for online analytical processing (OLAP). This paper is an initial guide to practical perfect hashing for OLAP. We identify several promising applications for PHFs in OLAP and survey their current use in systems and research prototypes. We then evaluate existing PHF approaches and quantify their impact on query performance. Our results are encouraging: in a real OLAP system, PHFs achieve end-to-end speedups of 1.7X and 3.1X for join and aggregate queries, respectively. Nevertheless, there is room for improvement. Future approaches that simultaneously achieve low build time and high probe throughput could offer additional performance increases.

Extending the Mochi Methodology to Enable Dynamic HPC Data Services

M. Dorier, P. Carns, R. Ross, S. Snyder, R. Latham, A. Gueroudji, G. Amvrosiadis, C. Cranor, J. Soumagne.

5th Workshop on Extreme-Scale Storage and Analysis, May 2024.

High-performance computing (HPC) applications and workflows are increasingly making use of custom data services to complement traditional parallel file systems with fast transient data management capabilities tailored to application specific needs. In the Mochi project we provide methodologies and tools that enable rapid development of custom HPC data services, including a collection of composable software components that can be combined to build complex distributed data services. Our initial version of Mochi targeted data services deployed with static configurations with a fixed number of nodes and minimal fault tolerance. However, there is a growing need for dynamic services that can adapt while running in response to changing workloads and system conditions.

In this paper we present our work to extend the Mochi architecture to support the development of dynamic data services. We achieve this by providing new Mochi components that support unified bootstrapping and online reconfiguration, fault detection, monitoring, and consensus. We also provide a methodology for deriving service-wide resilience from the resilience of each of the service's components.

continued on page 21

continued from page 20

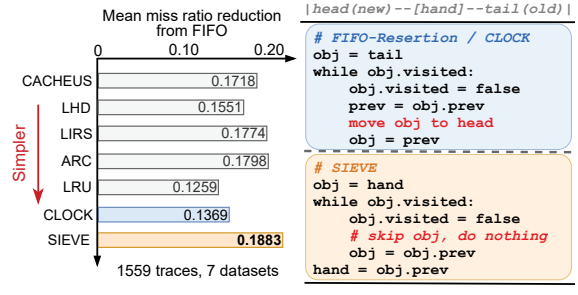
SIEVE is Simpler than LRU: An Efficient Turn-Key Eviction Algorithm for Web Caches

Yazhuo Zhang, Juncheng Yang, Yao Yue, Ymir Vigfusson, K. V. Rashmi

21st USENIX Symposium on Networked Systems Design and Implementation (NSDI'24), April 16–18, 2024. Santa Clara, CA.

COMMUNITY AWARD FOR BEST PAPER!

Caching is an indispensable technique for low-cost and fast data serving. The eviction algorithm, at the heart of a cache, has been primarily designed to maximize efficiency—reducing the cache miss ratio. Many eviction algorithms have been designed in the past decades. However, they all trade off throughput, simplicity, or both for higher efficiency. Such a compromise often hinders adoption in production systems. This work presents SIEVE, an algorithm that is simpler than LRU and provides better than state-of-the-art efficiency and scalability for web cache workloads. We implemented SIEVE in five production cache libraries, requiring fewer than 20 lines of code changes on average. Our evaluation on 1559 cache traces from 7 sources shows that SIEVE achieves up to 63.2% lower miss ratio than ARC. Moreover, SIEVE has a lower miss ratio than 9 state-of-the-art algorithms on more than 45% of the 1559 traces, while the next best algorithm only has a lower miss ratio on 15%. SIEVE's simplicity comes with superior scalability as cache hits require no locking. Our prototype achieves twice the throughput of an optimized 16-thread LRU implementation. SIEVE is more than an eviction algorithm; it can be used as a cache primitive to build advanced eviction algorithms just like FIFO and LRU.



SIEVE is simple and efficient. The code snippet shows how FIFOReinsertion and SIEVE find eviction candidates. Minor code changes convert FIFO-Reinsertion to SIEVE, unleashing lower miss ratios than state-of-the-art algorithms.

Dear User-Defined Functions, Inlining isn't working out so great for us. Let's try batching to make our relationship work. Sincerely, SQL

Kai Franz, Samuel Arch, Denis Hirn, Torsten Grust, Todd C. Mowry, Andrew Pavlo.

Conference on Innovative Data Systems Research (CIDR 2024), Chaminade, CA, USA, January 14-17, 2024.

SQL's user-defined functions (UDFs) allow developers to express complex computation using procedural logic. But UDFs have been the bane of database management systems (DBMSs) for decades because they inhibit optimization opportunities, potentially slowing down queries significantly. In response, batching and inlining techniques have been proposed to enable effective query optimization of UDF calls within SQL. Inlining is now available in a major commercial DBMS. But the trade-offs between both approaches on modern DBMSs remain unclear.

We evaluate and compare UDF batching and inlining on enterprise and open-source DBMSs using a state-of-the-art UDF-centric workload. We observe the surprising result that although inlining is better on simple UDFs, batching outperforms inlining by up to 93.4× for more complex UDFs because it makes it easier for a DBMS's query optimizer to decorrelate subqueries. We propose a hybrid approach that chooses batching or inlining to achieve the best performance.

Baleen: ML Admission & Prefetching for Flash Caches

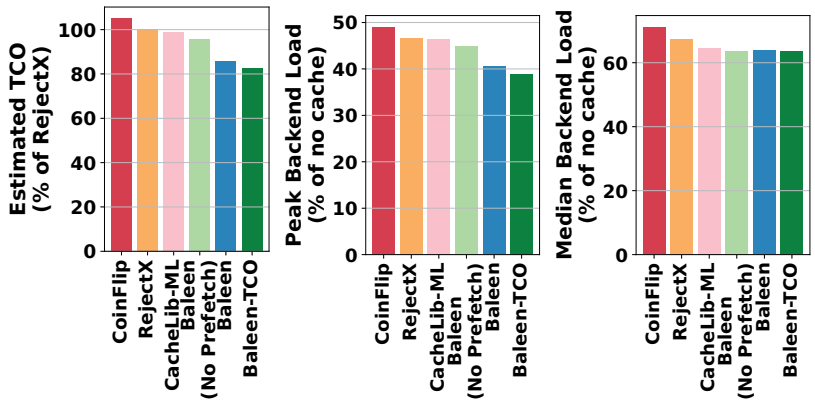
Daniel Lin-Kit Wong, Hao Wu, Carson Molder, Sathya Gunasekar, Jimmy Lu, Snehal Khandkar, Abhinav Sharma, Daniel S. Berger, Nathan Beckmann, Gregory R. Ganger

22nd USENIX Conference on File and Storage Technologies (FAST'24), Feb. 27–29, 2024, Santa Clara, CA.

Flash caches are used to reduce peak backend load for throughput-constrained data center services, reducing the total number of backend servers required. Bulk storage systems are a large-scale example, backed by high-capacity but low- throughput hard disks, and using flash caches to provide a more cost-effective storage layer underlying everything from blobstores to data warehouses. However, flash caches must address the limited write endurance of flash by limiting the long-term average flash write rate to avoid premature wearout. To do so, most flash caches must use admission policies to filter cache insertions and maximize the workload-reduction value of each flash write. The Baleen flash cache uses coordinated ML admission and prefetching to reduce peak backend load. After learning painful lessons with our early ML policy attempts, we exploit a new cache residency model (which we call episodes) to guide model training. We focus on optimizing for an end-to-end system metric (Disk-head Time) that measures backend load more accurately than IO miss rate or byte miss rate. Evaluation using Meta traces from seven storage clusters shows that Baleen reduces Peak Disk-head Time (and hence the number of backend

continued on page 22

continued from page 21



Baleen-TCO reduces (estimated) TCO by 17% and peak load by 16% over the best baseline on 7 Meta traces by choosing the optimal flash write rate. IO and byte miss rates were reduced by 14% and 2% (Suppl A.1). For the default flash write rate, Baleen reduces peak load by 12% over the best baseline.

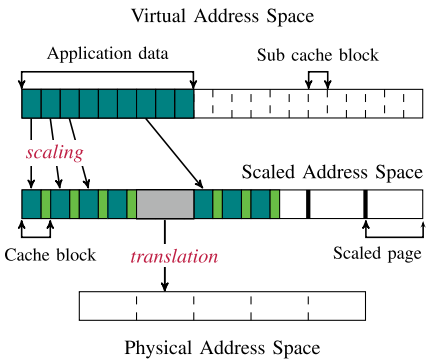
hard disks required) by 12% over state-of-the-art policies for a fixed flash write rate constraint. Baleen- TCO, which chooses an optimal flash write rate, reduces our estimated total cost of ownership (TCO) by 18%.

Address Scaling: Architectural Support for Fine-Grained Thread-Safe Metadata Management

Deepanjali Mishra, Konstantinos Kanellopoulos, Ashish Panwar, Akshitha Sriraman, Vivek Seshadri, Onur Mutlu, Todd C. Mowry

IEEE Computer Architecture Letters, Volume: 23, Issue: 1, Jan.-June 2024.

In recent decades, software systems have grown significantly in size and complexity. As a result, such systems are more prone to bugs which can cause performance and correctness challenges. Using run-time monitoring tools is one approach to mitigate these challenges. However, these tools maintain metadata for every byte of application data they monitor, which precipitates performance overheads from additional metadata accesses. We propose Address Scaling, a new hardware framework that performs fine-grained metadata management to reduce metadata access overheads in run-time monitoring tools. Our mechanism is based on the observation that different run-time monitoring tools maintain metadata at varied granularities. Our key insight is to maintain the data and its corresponding metadata within the same cache line, to preserve locality. Address Scaling improves the performance of Memcheck, a dynamic monitoring tool that detects memory-related errors, by 3.55×and 6.58×for sequential and random memory access



Conceptual design of Address Scaling. The dotted lines in the virtual address space indicate sub-cache-line boundaries. Each cache line in the scaled address space contains a sub-cache-line and metadata associated with it.

patterns respectively, compared to the state-of-the-art systems that store the metadata in a memory region that is separate from the data.

UDIR: Towards a Unified Compiler Framework for Reconfigurable Dataflow Architectures

Nikhil Agarwal, Mitchell Fream, Souradip Ghosh, Brian C. Schwedock, Nathan Beckmann

IEEE Computer Architecture Letters (Volume: 23, Issue: 1, Jan.-June 2024).

Specialized hardware accelerators have gained traction as a means to improve energy efficiency over inefficient von Neumann cores. However, as specialized hardware is limited to a few applications, there is increasing interest in programmable, non-von Neumann architectures to improve efficiency on a wider range of programs. Reconfigurable dataflow architectures are a promising design, but the design space is fragmented and, in particular, existing compiler and software stacks are ad hoc and hard to use. Without a robust, mature software ecosystem, RDAs lose much of their advantage over specialized hardware.

This paper proposes a unifying dataflow intermediate representation (UDIR) for reconfigurable dataflow compilers. Popular von Neumann compiler representations are inadequate for dataflow architectures because they do not represent the dataflow control paradigm, which is the target of many common compiler analyses and optimizations. UDIR introduces contexts to break regions of instruction reuse in programs. Contexts generalize prior dataflow control paradigms, representing where in the program tokens must be synchronized. We evaluate UDIR on four prior dataflow architectures, providing simple rewrite rules to lower UDIR to their respective machine-specific representations, and demonstrate

continued on page 23

continued from page 22

a case study of using UDIR to optimize memory ordering.

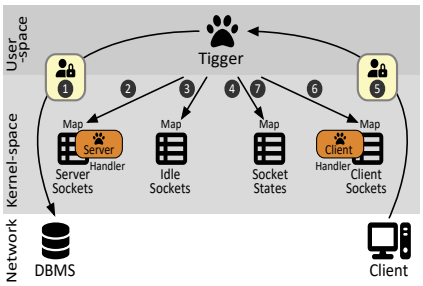
Tigger: A Database Proxy That Bounces With User-Bypass

Matthew Butrovich, Karthik Ramanathan, John Rollinson, Wan Shen Lim, William Zhang, Justine Sherry, Andrew Pavlo

Proceedings of the VLDB Endowment, Vol. 16, No. 11, 2023.

Developers often deploy database-specific network proxies wherebyapplications connect transparently to the proxy instead of directly connecting to the database management system (DBMS). This indirection improves system performance through connection pooling, load balancing, and other DBMS-specific optimizations. Instead of simply forwarding packets, these proxies implement DBMS protocol logic (i.e., at the application layer) to achieve this behavior. Consequently, existing proxies are user-space applications that process requests as they arrive on network sockets and forward them to the appropriate destinations. This approach incurs inefficiencies as the kernel repeatedly copies buffers between user-space and kernel-space, and the associated system calls add CPU overhead.

This paper presents user-bypass, a technique to eliminate these overheads by leveraging modern operating system features that support custom code execution. User-bypass pushes application logic into kernel-space via Linux's eBPF infrastructure. To demonstrate its benefits,we implemented Tigger, a PostgreSQL-compatible DBMS proxy using user-bypass to eliminate the overheads of traditional proxy design. We compare Tigger's performance against other state-of-the-art proxies widely used in real-world deployments. Our experiments show that Tigger outperforms other proxies — in one scenario achieving both the lowest transaction latencies (up to 29% reduction) and lowest CPU utilization (up to 42% reduction). The results



Tigger's User-Bypass Architecture – Tigger's hybrid design constrains user-space and kernel-space (i.e., eBPF programs and maps) components. After Tigger loads its handlers (i.e., Client and Server) and maps into kernelspace, 1) its user-space component opens and authenticates connections to the back-end DBMS for pooling. Next, 2) Tigger adds the server sockets to ServerSocketsMap. This step ensures that the Server handler runs whenever a DBMS socket buffer is ready for processing. With the back-end socket ready to accept queries, 3) Tigger adds it to the stack map of idle sockets. 4) Tigger resets the state metadata associated with the socket. Upon a new connection request, 5) the client authenticates with Tigger's user-space component. 6) Tigger adds the client's socket to ClientSocketsMap. The Client handler will now execute on buffer activity from a front-end connection. Lastly, 7) Tigger resets the metadata associated with the client socket stores in SocketStatesMap.

show that user-bypass implementations like Tigger are well-suited to DBMS proxies' unique requirements.

Rethinking the Encoding of Integers for Scans on Skewed Data

Martin Prammer, Jignesh M. Patel

Proc. ACM Manag. Data, Vol. 1, No. 4 (SIGMOD), Article 257. Dec. 2023.

Bit-parallel scanning techniques are characterized by their ability to accelerate compute through the process known as early pruning. Early pruning techniques iterate over the bits of each value, searching for opportunities to safely prune compute early, before processing each data value in its entirety. However, because of this iterative evaluation, the effectiveness of early pruning depends on the relative position of bits that can be used for pruning within each value. Due to this behavior, bit-parallel techniques have faced significant challenges when processing skewed data, especially when values contain many leading zeroes. This problem is further amplified by the inherent trade-off that bit-parallel techniques make between columnar scan and fetch performance: a storage layer that supports early pruning requires multiple memory accesses to fetch a single value. Thus, in the case of skewed data, bit-parallel techniques increase fetch latency without significantly improving scan performance when compared to baseline columnar implementations.

To remedy this shortcoming, we transform the values in bit-parallel columns using novel encodings. We propose the concept of forward encodings: a family of encodings that shift pruning-relevant bits closer to the most significant bit. Using this concept, we propose two particular encodings: the Data Forward Encoding and the Extended Data Forward Encoding. We demonstrate the impact of these encodings using multiple real-world datasets. Across these datasets, forward encodings improve the current state-of-the-art bit-parallel technique's scan and fetch performance in many cases by 1.4x and 1.3x, respectively.

FIFO Queues Are All You Need for Cache Eviction

Juncheng Yang, Yazhuo Zhang, Ziyue Qiu, Yao Yue, Rashmi Vinayak

SOSP '23: Proceedings of the 29th Symposium on Operating Systems Principles, October 2023. Koblenz, Germany.

As a cache eviction algorithm, FIFO has a lot of attractive properties, such as simplicity, speed, scalability, and flash-friendliness. The most prominent criticism of FIFO is its low efficiency (high miss ratio).

continued on page 24

continued from page 23

In this work, we demonstrate a simple, scalable FIFO-based algorithm with three static queues (S3-FIFO). Evaluated on 6594 cache traces from 14 datasets, we show that S3-FIFO has lower miss ratios than state-of-the-art algorithms across traces. Moreover, S3-FIFO's efficiency is robust --- it has the lowest mean miss ratio on 10 of the 14 datasets. FIFO queues enable S3-FIFO to achieve good scalability with 6× higher throughput compared to optimized LRU at 16 threads.

Our insight is that most objects in skewed workloads will only be accessed once in a short window, so it is critical to evict them early (also called quick demotion). The key of S3-FIFO is a small FIFO queue that filters out most objects from entering the main cache, which provides a guaranteed demotion speed and high demotion precision.

Memento: Architectural Support for Ephemeral Memory Management in Serverless Environments

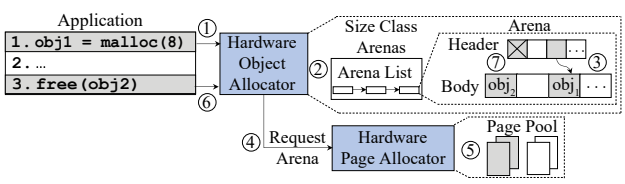
Ziqi Wang, Kaiyang Zhao, Pei Li, Andrew Jacob, Michael Kozuch, Todd Mowry, Dimitrios Skarlatos

MICRO '23: Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture. October 2023. Toronto, Canada.

Serverless computing is an increasingly attractive paradigm in the cloud due to its ease of use and fine-grained pay-for-what-you-use billing. However, serverless computing poses new challenges to system design due to its short-lived function execution model. Our detailed analysis reveals that memory management is responsible for a major amount of function execution cycles. This is because functions pay the full critical-path costs of memory management in both user space and the operating system without the opportunity to amortize these costs over their short lifetimes.

David Bonnie (LANL) and Ziyue Qiu discuss her work on "FrozenHot Cache: Rethinking Cache Management for Modern Hardware" at a PDL Retreat poster session.

To address this problem, we propose Memento, a new hardware-centric memory management design based upon our insights that memory allocations in serverless functions are typically small, and either quickly freed after allocation or freed when the function exits. Memento alleviates the overheads of serverless memory management by introducing two key mechanisms: (i) a hardware object allocator that performs in-cache memory allocation and free operations based on arenas, and (ii) a hardware page allocator that manages a small pool of physical pages used to replenish arenas of the object allocator. Together these mechanisms alleviate memory management overheads and bypass costly user space and kernel operations. Memento naturally integrates with existing software stacks through a set of ISA extensions that enable seamless integration with multiple languages runtimes. Finally, Memento leverages the newly exposed memory allocation semantics in hardware to introduce a main memory bypass mechanism and avoid unnecessary DRAM accesses for newly allocated objects.



Memento's memory management workflow.

We evaluate Memento with full-system simulations across a diverse set of containerized serverless workloads and language runtimes. The results show that Memento achieves function execution speedups ranging between 8–28% and 16% on average. Furthermore, Memento hardware allocators and main memory bypass mechanisms drastically reduce main memory traffic by 30% on average. The combined effects of Memento reduce the pricing cost of function execution by 29%. Finally, we demonstrate the applicability of Memento beyond functions, to major serverless platform operations and long-running data processing applications.

Simple Adaptive Query Processing vs. Learned Query Optimizers: Observations and Analysis

Yunjia Zhang, Yannis Chronis, Jignesh M. Patel, Theodoros Rekatsinas

Proceedings of the VLDB Endowment, Vol. 16, No. 11.

There have been many decades of work on optimizing query processing in database management systems. Recently, modern machine learning (ML), and specifically reinforcement learning (RL), has gained increased attention as a means to develop a query optimizer (QO). In this work, we take a closer look at two recent state-of-the-art (SOTA) RL-based QO methods to better understand their behavior.We find that these RL-based methods do not generalize as well as it seems at first

continued on page 25

continued from page 24

glance. So, how do SOTA RL-based QOs compare to a simple, modern, adaptive query processing approach? To answer this question, we choose two simple adaptive query processing techniques and implemented them in PostgreSQL. The first adapts an individual join operation on-the-fly and switches between a Nested Loop Join algorithm and a Hash Join algorithm to avoid sub-optimal join algorithm decisions. The second is a technique called Lookahead Information Passing (LIP), in which adaptive semijoin techniques are used to make a pipeline of join operations execute efficiently. To our surprise, we find that this simple adaptive query processing approach is not only competitive to the SOTA RL-based approaches but, in some cases, outperforms the RL-based approaches. The adaptive approach is also appealing because it does not require an expensive training step, and it is fully interpretable compared to the RL-based QO approaches. Further, the adaptive method works across complex query constructs that RL-based QO methods currently cannot optimize.

Peeling Back the Carbon Curtain: Carbon Optimization Challenges in Cloud Computing

Jaylen Wang, Udit Gupta, Akshitha Sriraman

HotCarbon 2023. July 9, 2023, Boston, MA, USA.

The increasing carbon emissions from cloud computing requires new methods to reduce its environmental impact. We explore extending data center server lifetimes to reduce embodied carbon emissions (from hardware manufacturing), rather than operational (from running hardware). Our experiments are the first to analyze a data center application's end-to-end performance on different server generations, to reveal that older hardware can preserve performance in certain conditions (e.g., low load).

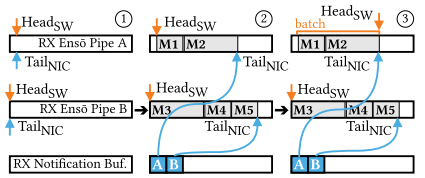
Our observations show the need for a carbon-aware data center scheduler that schedules on older hardware when suitable. However, quantifying such a scheduler's carbon savings is challenging today due to the lack of practical carbon measurement metrics/tools. We identify gaps in current methods for measuring operational and embodied carbon and call upon the broader systems research community to take action and conduct research that can pave the way for future carbon footprint analysis in systems.

Ensō: A Streaming Interface for NIC-Application Communication

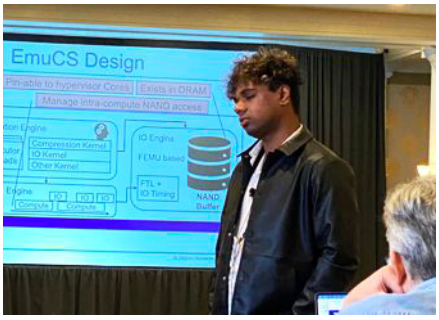
Hugo Sadok, Nirav Atre, Zhipeng Zhao, Daniel S. Berger, James C. Hoe, Aurojit Panda, Justine Sherry, Ren Wang

17th USENIX Symp. on Operating Systems Design and Implementation (OSDI). July 10–12, 2023. Boston, MA.

Today, most communication between the NIC and software involves exchanging fixed-size packet buffers. This packetized interface was designed for an era when NICs implemented few offloads and software implemented the logic for translating between application data and packets. However, both NICs and networked software have evolved: modern NICs implement hardware offloads, e.g., TSO, LRO, and serialization offloads that can more efficiently translate between application data and packets. Furthermore, modern software increasingly batches network I/O to reduce overheads. These changes have led to a mismatch between the packetized interface, which assumes that the NIC and software exchange fixed-size buffers, and the features provided by modern NICs and used by modern software. This incongruence between interface and data adds software complexity and I/O overheads, which in turn limits communication performance.



Steps to receive batches of messages in two Ensō Pipes.



Nj Muckherjee considers an industry guest's question following his talk on "Near Data Block Processing Using Computational Storage" at the 2023 PDL Retreat.

This paper proposes Ensō, a new streaming NIC-to-software interface designed to better support how NICs and software interact today. At its core, Ensō eschews fixed-size buffers, and instead structures communication as a stream that can be used to send arbitrary data sizes. We show that this change reduces software overheads, reduces PCIe bandwidth requirements, and leads to fewer cache misses. These improvements allow an Ensō-based NIC to saturate a 100 Gbps link with minimum-sized packets (forwarding at 148.8 Mpps) using a single core, improve throughput for high-performance network applications by 1.5–6×, and reduce latency by up to 43%.

continued on page 26

continued from page 25

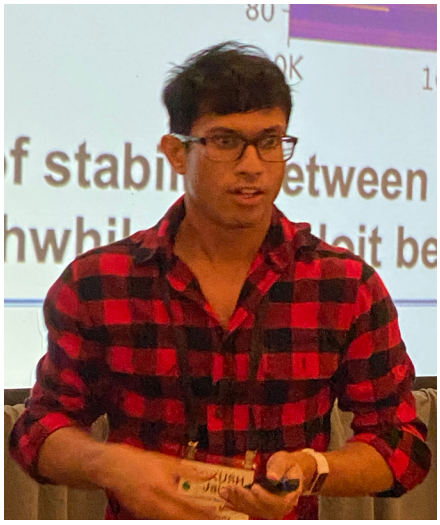
Runahead A*: Speculative Parallelism for A* with Slow Expansions

Mohammad Bakhshalipour, Mohamad Qadri, Dominic Guri, Seyed Borna, Ehsani, Maxim Likhachev, Phillip B. Gibbons

ICAPS 2023, Prague, Czech Republic, July 8-13, 2023.

A* suffers from limited parallelism. The maximum level of traditional parallelism in A* is the same as the degree of the search graph nodes, which is too small in many applications. As such, A* cannot fully leverage the multithreading capabilities of modern processors. In this paper, we go beyond traditional parallelism and introduce speculative parallelism for A*. We observe that A*'s node expansions exhibit predictable patterns in applications like path planning. Based on this observation, we propose Runahead A* (RA*). When a node is being expanded, RA* predicts future likely-to-be-expanded nodes, performs their corresponding computation on separate threads, and memoizes the computation results. Later when a predicted node is selected for expansion, rather than performing its computation, the memoized results are used, saving significant time in slow-expansion applications. We study five applications of A*. We show that when its prediction accuracy is high, RA* offers significant speedup over vanilla A* for slow-expansion applications. With 16 threads, RA*'s speedup for such applications ranges from 3.1× to 14.1×. We also study and provide insight into when, why, and to what extent node expansions are predictable. We provide an implementation of RA* at: https://github.com/cmu-roboarch/runahead-astar/

Juncheng Yang, starting soon as an Assistant Professor at Harvard, with his thesis committee: Top: Vijay Chidambaram (UT Austin), Ion Stoica (UC Berkeley); Bottom: Phillip Gibbons, JY, Rashmi Vinayak (chair), and Greg Ganger.



Ankush Jain presents his research on "Optimizing Adaptive Physics Apps with Better Work Placement" at the 2023 PDL Retreat.

Sia: Heterogeneity-aware, Goodput-optimized ML-cluster Scheduling

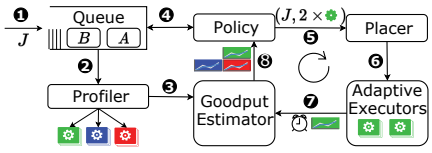
Suhas Jayaram Subramanya, Daiyaan Arfeen, Shouxu Lin, Aurick Qiao, Zhihao Jia, and Gregory R. Ganger

ACM SIGOPS 29th Symp. on Operating Systems Principles (SOSP '23), Oct.23–26, 2023, Koblenz, Germany.

The Sia* scheduler efficiently assigns heterogeneous deep learning (DL) cluster resources to elastic resource-adaptive jobs. Although some recent schedulers address one aspect or another (e.g., heterogeneity or resource-adaptivity), none addresses all and most scale poorly to large clusters and/or heavy workloads even without the full complexity of the combined scheduling problem. Sia introduces a new scheduling formulation that can scale to the search-space sizes and intentionally match jobs and their configurations to GPU types and counts, while adapting to changes in cluster load and job mix over time. Sia also introduces a low-profiling overhead approach to bootstrapping (for each new job) throughput models used to evaluate possible resource assignments, and it is the first cluster scheduler to support elastic scaling of hybrid parallel jobs.

Extensive evaluations show that Sia outperforms state-of-the-art schedulers. For example, even on relatively small 44- to 64-GPU clusters with a mix of three GPU types, Sia reduces average job completion time (JCT) by 30–93%, 99th percentile JCT and makespan by 28–95%, and GPU hours used by 12–55% for workloads derived from 3 real-world environments. Additional experiments demonstrate that Sia scales to at least 2000-GPU clusters, provides improved fairness, and is not over-sensitive to scheduler parameter settings.

*In Egyptian mythology, Sia is the god of perception/intelligence, not to be confused with the popular music artist.



Lifecyle of a job under Sia. After a job is submitted, it is profiled once on each GPU type for a few batchsizes. Upon receiving an allocation, the job begins a cycle of continuous optimization (steps 5-8) for the remainder of its life in the cluster. Policy continuously optimizes allocations for the job, while Goodput Estimator provides up-to-date performance and gradient statistics to Policy to aid in decision making.

continued from page 9

becomes less reliable with both lifetime and IO accesses. Their research focuses on how to reduce IO to enable longer storage-device lifetimes, thus enabling sustainable datacenter storage.

Rising Stars is an intensive workshop for graduate students and postdocs with historically marginalized or underrepresented genders who are interested in pursuing academic careers in electrical engineering, computer science, and artificial intelligence and decision-making. Launched at MIT in 2012, the annual event has since been hosted at the University of California at Berkeley, Carnegie Mellon University, Stanford University, and the University of Illinois at Urbana-Champaign.

November 2023
Niraj Tolia Receives 2023 Parallel Data Lab Distinguished Alumni Award

Carnegie Mellon University's Parallel Data Lab (PDL) has recognized Niraj Tolia (BS '02, MS '03, Ph.D. '07), CEO and Co-Founder of Alcion, as the recipient of its 2023 Distinguished Alumni Award. PDL announced the honor at its 29th Annual Workshop and Retreat, held November 6-8 at the Omni Bedford Springs Resort in Bedford, PA.

Niraj received his Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University, where he was a part of the Parallel Data Lab. His research concentrated on distributed storage systems with an emphasis on deduplication and wide-area network data transfer.

November 2023
MICRO Hall of Famers!

Congratulations to CSD faculty members Todd Mowry and Nathan Beckmann for being inducted into the MICRO Hall of Fame, which recognizes authors who have had eight or more papers published in ACM SIGMICRO.

Attendees of the 29th PDL Retreat held at the Bedford Springs Resort, in Bedford, PA, November 6-8, 2023.