



PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • SUMMER 2022

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

RECENT PUBLICATIONS

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

Recent Publications	1
Director's Letter	2
Year in Review	4
PDL News & Awards	8
New PDL Faculty	11
Defenses & Proposals	12
PDL Alumni News	28

PDL CONSORTIUM MEMBERS

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

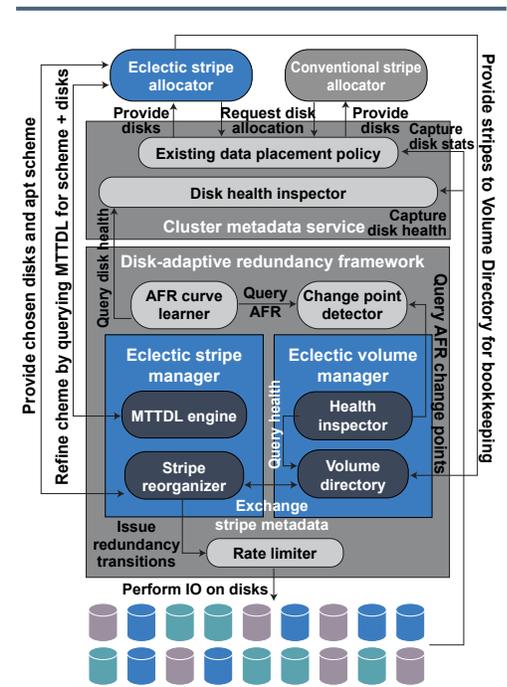
Tiger: Disk-Adaptive Redundancy Without Placement Restrictions

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

Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI), July 11–13, 2022. Carlsbad, CA, USA.

Large-scale cluster storage systems use redundancy (via erasure coding) to ensure data durability. Disk-adaptive redundancy—dynamically tailoring the redundancy scheme to observed disk failure rates—promises significant space and cost savings. Existing disk-adaptive redundancy systems, however, pose undesirable constraints on data placement, partitioning disks into subclusters that have homogeneous failure rates and forcing each erasure-coded stripe to be entirely placed on the disks within one subcluster. This design increases risk, by reducing intra-stripe diversity and being more susceptible to unanticipated changes in a make/model's failure rate, and only works for very large storage clusters fully committed to disk-adaptive redundancy.

Tiger is a new disk-adaptive redundancy system that efficiently avoids adoption-blocking placement constraints, while also providing higher space-savings and lower risk relative to prior designs. To do so, Tiger introduces the eclectic stripe, in which redundancy is tailored to the potentially-diverse failure rates of whichever disks are selected for storing that particular stripe. With eclectic stripes, pre-existing placement policies can be used while still enjoying the space-savings and robustness benefits of disk-adaptive redundancy. This paper introduces eclectic striping and Tiger's design, including a new mean-time-to-data-loss (MTTDL) approximation technique and new approaches for ensuring safe per-stripe settings given that failure rates of different devices change over time. In addition



Architecture of Tiger. The blue boxes correspond to Tiger's components. The gray boxes correspond to existing components in cluster storage system architecture and components present in existing disk-adaptive redundancy systems.

continued on page 5

FROM THE DIRECTOR'S CHAIR

GREG GANGER



At risk of jinxing things: we're back! We didn't get to do a PDL Retreat in Fall 2021, as we had hoped and planned, but we did an in-person PDL Visit Day in May 2022 and are fully on track for a full PDL Retreat in November 2022. We are very excited to get back to gathering in-person regularly to interact, share, brainstorm, and collaborate.

It has been an amazing year for PDL, on the research and student accomplishment fronts, despite the life challenges from which we are all emerging. Indeed, PDL researchers were recognized, over and over, with an unprecedented run of Best Paper awards (OSDI 2021, NSDI 2021, SOSP 2021, Eurosys 2022, ASPLOS 2022)... and Rashmi won the VMware Systems Award, George and team won an R&D100 award, and there were several other awards and nominations as well. Behind the awards, of course, has been great progress on many fronts and new projects/collaborations initiated. A highlight has been strong continued interaction with PDL sponsors, including cool guest lectures to PDL's storage systems and cloud classes and co-authored papers (including some of the awards) with us in the context of research collaborations. I will not try to cover all of the PDL progress across storage systems, database systems, ML systems, and data processing infrastructure—specifics can be found throughout the newsletter—but I will briefly highlight a few things.

I'll start with caching and storage systems, where great research results and awards have arisen from our collaborations with companies. As one example, the Kangaroo Flash cache design enables unprecedented efficiency for Flash caching of small objects... and was presented as Best Paper at SOSP 2021. As another, the SegCache memory cache designs enables unprecedented scalability for in-memory key-value caching... and was presented as Best Paper at NSDI 2021. The disk-adaptive redundancy project has created a new flexible way to work with existing data-placement policies and achieve even bigger capacity-reduction benefits for cluster storage systems; the flexible HPC storage project's microservice-based approach received an R&D100 award; there are cool new software designs for emerging device interfaces (e.g., ZNS) and for emerging smart storage devices; etc. We thank our PDL sponsor companies who have enabled (and collaborated on) much of the research mentioned above by allowing us to experiment with real devices, workload traces, and failure logs!

We also continue our extensive work in large-scale data processing systems, including systems for ML and schedulers for analytics clusters. For example, the Pollux scheduler for shared GPU clusters co-adaptively makes resource assignment decisions and job configuration decisions (e.g., batch size) for DNN-training jobs to achieve unprecedented ef-

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.

FACULTY

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

George Amvrosiadis	Gauri Joshi
David Andersen	Todd Mowry
Nathan Beckmann	David O'Hallaron
Chuck Cranor	Andy Pavlo
Lorrie Cranor	Majd Sakr
Christos Faloutsos	M. Satyanarayanan
Phil Gibbons	Dimitrios Skarlatos
Mor Harchol-Balter	Akshitha Sriraman
Zhihao Jia	Rashmi Vinayak

STAFF MEMBERS

Bill Courtright, 412•268•5485
(PDL Executive Director) wcourtright@cmu.edu
Karen Lindenfelser, 412•268•6716
(PDL Administrative Manager) karen@ece.cmu.edu
Jason Boles
Joan Digney
Chad Dougherty
Mitch Franzos
Baljit Singh

**VISITING RESEARCHERS &
POST DOCS**

Andrew Crotty

GRADUATE STUDENTS

Nikhil Agarwal	Wan Shen Lim
Daiyaan Arfeen	Shouxu Lin
Nikhil Arora	Elliot Lockerman
Sanjith Athlur	Hao Yang Lu
Nirav Atre	Francisco Maturana
Mohammad Bakhshalipour	Sara McAllister
Matt Butrovich	Nj Mukherjee
Zixu Chen	Hojin Park
Neville Chima	Deepayan Patra
Yae Jee Cho	Xavier Ramos
Val Choung	Sumanth Rao
Andrew Chung	Brian Schwedock
Pratik Fegade	Shalini Shukla
Samarth Gupta	Suhas J Subramanya
Eeshwar Gurushankar	Minh Truong
Travis Hance	Jaylen Wang
Garrison Hess	Sherry Wang
Ankush Jain	Yuanxin Wang
Jekyeom Jeon	Ziqi Wang
Ellango Jothimurugesan	Daniel Wong
Shantanu Kamath	Mike Xu
Akshath Karanam	Jason Yang
Thomas Kim	Jiuzhi Yu
Timothy Kim	Will Zhang
Jack Kosaian	Kaiyang Zhao
Joseph Koshakow	Yiwei Zhao
Michael Kuchnik	Giulio Zhou

UNDERGRADUATE STUDENTS

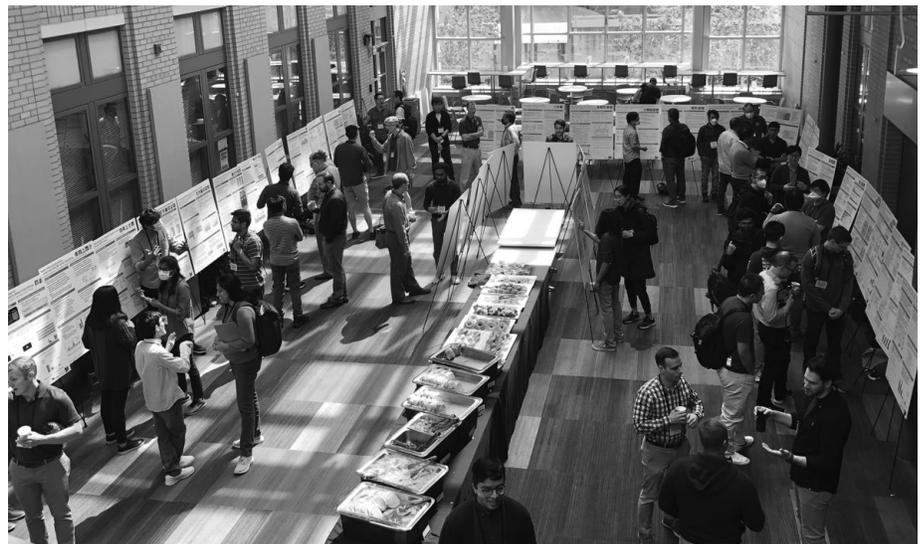
Jonathan Chiu

efficiency... and was presented as Best Paper at OSDI 2021. We are extending our recent work on how inter-job dependency information can be uncovered from provenance information and job logs to improve job shifting and scheduling that exploits intermittent resource availability (e.g., best effort resources and green energy). An emerging area of exploration is automated storage and caching configuration in cloud infrastructures to accommodate large-scale data processing within and across clouds. And, excitingly, broad aggregated activities are emerging among the research into simplifying, automating, and improving efficiency in big-data ML systems, both in collaboration with other CMU groups and with the Army's Pittsburgh-homed AI Integration Center (AI2C), as it looks to adopt and adapt key concepts into the new mechanisms they are building for AI development and use in the Army.

Database systems research continues strong, and the various approaches to automation are proving to be effective and impactful. The OtterTune service embodies blockbox approaches for database tuning and is part of a startup company, associated with a "music" release! The NoisePage DBMS embodies whitebox approaches, in which the DBMS is designed from the beginning to automatically adapt and is also emerging strong... and three researchers involved with it moved on to new faculty positions this summer!

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

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



PDL Consortium members, guests and CMU students, staff and faculty gather for the opening poster session at the 2022 Spring Visit Day.

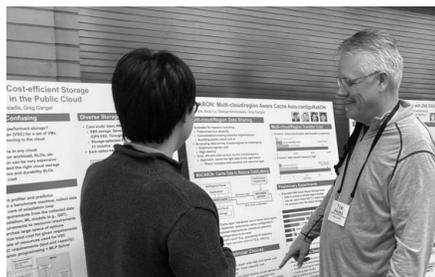
YEAR IN REVIEW

July 2022

- ❖ Nirav Atre gave his speaking skills talk on “SurgeProtector: Mitigating Temporal Algorithmic Complexity Attacks using Adversarial Scheduling.”
- ❖ Saurabh Kadekodi presented “Tiger: Disk-Adaptive Redundancy Without Placement Restrictions” at OSDI '22 in Carlsbad, CA.
- ❖ Benjamin Samuel Berg defended his PhD thesis research on “A Principled Approach to Parallel Job Scheduling.”
- ❖ Graham Gobieski defended his PhD thesis research on “Ultra-low-power, Energy-minimal Computer Architectures.”

June 2022

- ❖ Matthew Butrovich presented “Tastes Great! Less Filling! High Performance and Accurate Training Data Collection for Self-Driving Database Management Systems” at SIGMOD '22, in Philadelphia, PA.
- ❖ Brian C. Schwedock presented “täko: A Polymorphic Cache Hierarchy for General-Purpose Optimization of Data Movement” at ISCA '22, in New York, NY. Brian and his co-authors were nominated for best paper!
- ❖ Minh S. Q. Truong and Mohammad Bakhshalipour received an Apple Inc. Ph.D. Fellowship in Integrated Systems for 2021-2023. Minh interned this summer with



Hojin Park discusses his poster on “Mimir: Finding Cost-efficient Storage Configurations in the Public Cloud” with Tom Hicks of Microsoft Research.

the Neural Engine HW Architecture group at Apple Inc. under Jaewon Shin. Mohammad interned with Apple’s Platform Architecture team in both 2021 and 2022.

May 2022

- ❖ 23rd Annual Spring Visit Day and Consortium Speaker Series!
- ❖ Michael Kuchnik proposed his PhD thesis research on “Beyond Model Efficiency: Data Optimizations for Machine Learning Systems.”
- ❖ Thomas Sheek Kim proposed his PhD research topic “Leveraging Emerging Storage Technologies to Improve Performance, Reliability, and Availability.”

April 2022

- ❖ Francisco Maturana gave his speaking skills talk on “Convertible Codes: Efficient Conversion of Coded Data in Distributed Storage.”
- ❖ Sanjith Athlur and his co-authors won Best Paper at EuroSys'22 for their paper “Varuna: Scalable, Low-cost Training of Massive Deep Learning Models” in Rennes, France.

March 2022

- ❖ Yae Jee Cho, a member of Gauri Joshi’s team, presented “Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies” at the AISTATS '22 virtual conference.
- ❖ Todd Mowry’s 1992 Paper “Design and Evaluation of a Compiler Algorithm for Prefetching” was noted as Influential by ASPLOS'22.
- ❖ Ellango Jothimurugesan proposed his PhD research on “Large-scale Machine Learning over Streaming Data.”

February 2022

- ❖ Dimitrios Skarlatos and his co-authors won the Best Paper Award

at ASPLOS '22 for their paper “TMO: Transparent Memory Offloading in Datacenters” in Lausanne, Switzerland.

- ❖ Sol Boucher presented a defense of his PhD research on “Lightweight Preemptible Functions.”

January 2022

- ❖ Rashmi Vinayak received the 2021 VMware Systems Research Award for her work to enhance computer system reliability and efficiency by combining coding-theoretic algorithms, machine learning models and systems.
- ❖ Ellango Jothimurugesan gave his speaking skills talk on “DriftS-urf: Stable-State / Reactive-State Learning under Concept Drift.”
- ❖ Andrew Crotty presented “Are You Sure You Want to Use MMAP in Your Database Management System?” at CIDR '22 in Chaminade, CA.

December 2021

- ❖ Gauri Joshi’s team presented “Leveraging Spatial and Temporal Correlations in Sparsified Mean Estimation” at the virtual NeurIPS '21 conference.
- ❖ Graham Gobieski proposed his PhD research on “Ultra-low-power, Energy-minimal Computer Architectures.”

November 2021

- ❖ The PDL paper “Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis,” by Alexey Tumanov, Greg Ganger and co-authors won an ACM SoCC Test of Time Award.
- ❖ Jack Kosaian presented “Arithmetic-Intensity-Guided Fault Tolerance for Neural Network Inference on GPUs” at SC'21, in St. Louis, MO.
- ❖ Qing Zheng presented “DeltaFS: A Scalable No-Ground-Truth Filesystem For Massively-Parallel

continued on page 27

continued from page 1

to avoiding placement constraints, evaluation with logs from real-world clusters shows that Tiger provides better space-savings, less bursty IO for changing redundancy schemes, and better robustness (due to increased risk-diversity) than prior disk-adaptive redundancy designs.

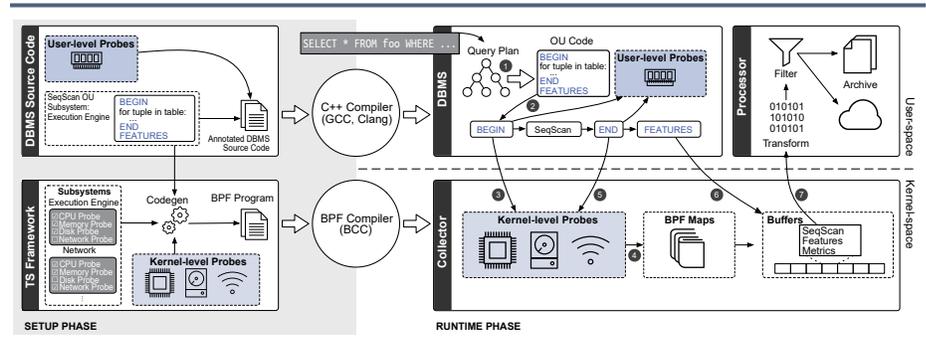
Tastes Great! Less Filling! High Performance and Accurate Training Data Collection for Self-Driving Database Management Systems

Matthew Butrovich, Wan Shen Lim, Lin Ma, John Rollinson, William Zhang, Yu Xia, Andrew Pavlo

SIGMOD '22, June 12–17, 2022, Philadelphia, PA, USA.

A self-driving database management system (DBMS) aims to configure, deploy, and optimize almost all aspects of itself automatically without human intervention or guidance. Achieving this high level of automation relies on machine learning (ML) models that predict how a DBMS will behave in different scenarios. This behavior encompasses all DBMS runtime operations, including query execution and maintenance tasks. These ML-based behavior models for a self-driving DBMS require low-level training data about a DBMS's internals. Such training data includes (1) features that describe the workload, environment, and DBMS configuration, and (2) both DBMS- and hardware-level metrics. But it is difficult to collect training data from a DBMS while it is running because it can introduce performance and measurement degradations that hinder the ML models' ability to predict the DBMS's behavior correctly.

We present the TScout (TS) framework for collecting training data from self-driving DBMSs. Our framework is an internal approach where developers annotate a DBMS's source code with hooks to monitor the system's behavior. TS then extracts these hooks and



TScout's architecture is split into two phases. In the Setup Phase, a developer annotates the DBMS's source code. Then, TS uses code generation to create a customized Collector for the Runtime Phase when the Collector retrieves metrics and the DBMS's OU input features to create training data. Finally, data are shipped to the Processor for archiving with other training data. The DBMS is deployed together with TS on the same server. [1] The DBMS executes the query's plan. [2] The thread executes the OU for the annotated sequential scan. [3] When the DBMS encounters the `BEGIN` marker for this OU, it triggers TS to enable metrics collection using its probes. [4] The Collector uses a BPF map to store a snapshot of probe results at the `BEGIN` marker. [5] After completion of the OU and triggering the `END` marker, the Collector retrieves this data from the BPF map, invokes the necessary probes again to get a current snapshot, computes the value for each metric, and stores the final results back in the BPF map. When the DBMS reaches the OU's `FEATURES` marker, [6] the Collector packages the features and metrics together into a sample data point and [7] sends it to the Processor via a perf ring buffer.

generates a kernel-level program (via Linux's BPF) that efficiently captures metrics from multiple sources (e.g., CPU performance counters, memory allocators). TS combines these metrics with internal DBMS state observations, generating training data for behavior models. We integrated TS in a PostgreSQL-compatible DBMS and measured its ability to collect training data for both OLTP and OLAP workloads. Our results show that TS generates training data for a deployed DBMS to train more accurate models than previous methods with only a 7% performance reduction.

täko : A Polymorphic Cache Hierarchy for General-Purpose Optimization of Data Movement

Brian C. Schwedock, Piratach Yoovidhya, Jennifer Seibert, Nathan Beckmann

ISCA '22, June 18–22, 2022, New York, NY, USA. Nominated for Best Paper!

Current systems hide data movement from software behind the load-store

interface. Software's inability to observe and respond to data movement is the root cause of many inefficiencies, including the growing fraction of execution time and energy devoted to data movement itself. Recent specialized memory-hierarchy designs prove that large data-movement savings are possible. However, these designs require custom hardware, raising a large barrier to their practical adoption.

This paper argues that the hardware-software interface is the problem, and custom hardware is often unnecessary with an expanded interface. The `täko` architecture lets software observe data movement and interpose when desired. Specifically, caches in `täko` can trigger software callbacks in response to misses, evictions, and writebacks. Callbacks run on reconfigurable data-flow engines placed near caches. Five case studies show that this interface covers a wide range of data-movement features and optimizations. Microarchitecturally, `täko` is similar to recent near-data computing designs, adding $\approx 5\%$ area to a baseline multicore. `täko` improves performance by $1.4\times$ – $4.2\times$,

continued on page 6

RECENT PUBLICATIONS

continued from page 5

similar to prior custom hardware designs, and comes within 1.8% of an idealized implementation.

Mimir: Finding Cost-efficient Storage Configurations in the Public Cloud

Hojin Park, Gregory R. Ganger, George Amvrosiadis

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-22-102, February 2022.

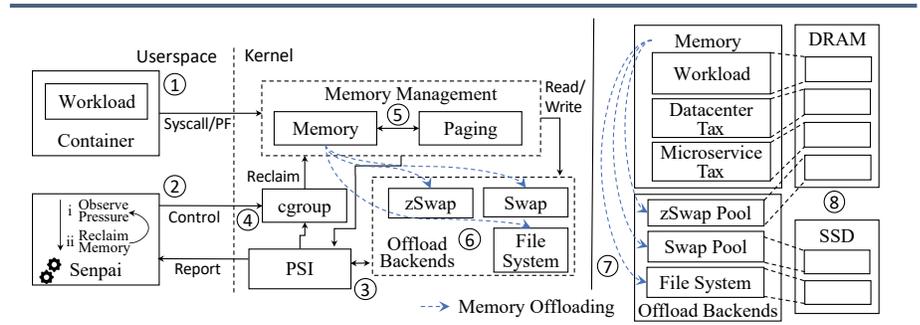
Public cloud providers offer a diverse collection of block storage options with different costs and performance SLAs. As a consequence, it is difficult to select the right allocations for storage backends when moving data-heavy applications to the cloud. Mimir is a tool for automatically finding a cost-efficient virtual storage cluster (VSC) configuration for a customer's storage workload and performance requirements. Importantly, since no single allocation type is best for all workloads, Mimir considers all allocation types and even heterogeneous mixes of them. In our experiments, compared to state-of-the-art approaches that consider only one allocation type, Mimir finds VSC configurations that reduce cost by up to 81% for substantial storage workloads.

TMO: Transparent Memory Offloading in Datacenters

Johannes Weiner, Niket Agarwal, Dan Schatzberg, Leon Yang, Hao Wang, Blaise Sanouillet, Bikash Sharma, Tejun Heo, Mayank Jain, Chunqiang Tang, Dimitrios Skarlatos

ASPLOS '22, February 28 – March 4, 2022, Lausanne, Switzerland. BEST PAPER AWARD!

The unrelenting growth of the memory needs of emerging datacenter applications, along with ever increasing cost and volatility of DRAM prices, has led to DRAM being a major infrastructure expense. Alternative technologies, such as NVMe SSDs and upcoming



Overview of the TMO architecture (left) and the memory and storage layout (right).

NVM devices, offer higher capacity than DRAM at a fraction of the cost and power. One promising approach is to transparently offload colder memory to cheaper memory technologies via kernel or hypervisor techniques. The key challenge, however, is to develop a datacenter-scale solution that is robust in dealing with diverse workloads and large performance variance of different offload devices such as compressed memory, SSD, and NVM.

This paper presents TMO, Meta's transparent memory offloading solution for heterogeneous datacenter environments. TMO introduces a new Linux kernel mechanism that directly measures in realtime the lost work due to resource shortage across CPU, memory, and I/O. Guided by this information and without any prior application knowledge, TMO automatically adjusts how much memory to offload to heterogeneous devices (e.g., compressed memory or SSD) according to the device's performance characteristics and the application's sensitivity to memory-access slowdown. TMO holistically identifies offloading opportunities from not only the application containers but also the sidecar containers that provide infrastructure-level functions. To maximize memory savings, TMO targets both anonymous memory and file cache, and balances the swap-in rate of anonymous memory and the reload rate of file pages that were recently evicted from the file cache.

TMO has been running in produc-

tion for more than a year, and has saved between 20-32% of the total memory across millions of servers in our large datacenter fleet. We have successfully upstreamed TMO into the Linux kernel.

Client Selection in Federated Learning: Convergence Analysis and Power-of-Choice Selection Strategies

Yae Jee Cho, Jianyu Wang, Gauri Joshi

International Conference on Artificial Intelligence and Statistics (AISTATS), March 2022.

Federated learning is a distributed optimization paradigm that enables a large number of resource-limited client nodes to cooperatively train a model without data sharing. Several works have analyzed the convergence of federated learning by accounting of data heterogeneity, communication and computation limitations, and partial client participation. However, they assume unbiased client participation, where clients are selected at random or in proportion of their data sizes. In this paper, we present the first convergence analysis of federated optimization for biased client selection strategies, and quantify how the selection bias affects convergence speed. We reveal that biasing client selection towards clients with higher local loss achieves faster error convergence. Using this insight, we propose POWER-OF-CHOICE, a communication- and computation-efficient client selection framework

continued on page 7

continued from page 6

that can flexibly span the trade-off between convergence speed and solution bias. Our experiments demonstrate that POWER-OF-CHOICE strategies converge up to 3 faster and give 10% higher test accuracy than the baseline random selection.

Adapting the RACER Architecture to Integrate Improved In-ReRAM Logic Primitives

Minh S. O. Truong, Liting Shen, Alexander Glass, Alison Hoffmann, L. Richard Carley, James A. Bain, Saugata Ghose.

IEEE Journal on Emerging and Selected Topics in Circuits and Systems, Early Access, 12 May 2022.

Modern computing applications based upon machine learning can incur significant data movement overheads in state-of-the-art computers. Resistive-memory-based processing-using-memory (PUM) can mitigate this data movement by instead performing computation in situ (i.e., directly within memory cells), but device-level limitations restrict the practicality and/or performance of many PUM architecture proposals. The RACER architecture overcomes these limitations, by proposing efficient peripheral circuitry and the concept of bit-pipelining to enable high-performance, high-efficiency computation using small memory tiles. In this work, we extend RACER to adapt easily to different PUM logic families, by (1) modifying the device access circuitry to support a wide range of logic families, (2) evaluating three logic families proposed by prior work, and (3) proposing and evaluating a new logic family called OSCAR that significantly relaxes the switching voltage constraints required to perform logic with resistive memory devices. We show that the modified RACER architecture, using the OSCAR logic family, can enable practical PUM on real ReRAM devices while improving

performance and energy savings by 30% and 37%, respectively, over the original RACER work.

Varuna: Scalable, Low-cost Training of Massive Deep Learning Models

Sanjith Athlur, Nitika Saran, Muthian Sivathanu, Ramachandran Ramjee, Nipun Kwatra

EuroSys '22, April 5-8, 2022, Rennes, France. BEST PAPER AWARD!

Systems for training massive deep learning models (billions of parameters) today assume and require specialized “hyperclusters”: hundreds or thousands of GPUs wired with specialized high-bandwidth interconnects such as NV-Link and Infiniband. Besides being expensive, such dependence on hyperclusters and custom high-speed inter-connects limits the size of such clusters, creating (a) scalability limits on job parallelism; (b) resource fragmentation across hyperclusters.

In this paper, we present Varuna a new system that enables training massive deep learning models on commodity networking. Varuna makes thrifty use of networking resources and automatically configures the user’s training job to efficiently use any given set of resources. Therefore, Varuna is able to leverage “low-priority” VMs that cost about 5x cheaper than dedicated GPUs, thus significantly reducing the cost of training massive models. We demonstrate the efficacy of Varuna by training massive models, including a 200 billion parameter model, on 5x cheaper “spot VMs”, while maintaining high training throughput. Varuna improves end-to-end training time for language models like BERT and GPT-2 by up to 18x compared to other

model-parallel approaches and up to 26% compared to other pipeline parallel approaches on commodity VMs. The code for Varuna is available at <https://github.com/microsoft/varuna>.

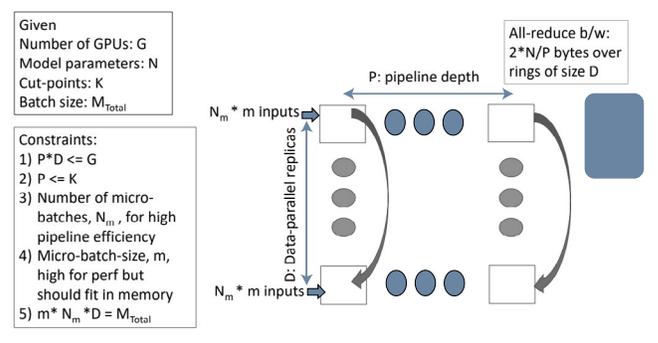
RAIZN: Redundant Array of Independent Zoned Namespaces

Thomas Kim, George Amvrosiadis, Jekyeom Jeon, Huaicheng Li, David G. Andersen, Greg Ganger, Michael Kaminsky, Matias Bjørling

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-22-101, January 2022.

Zoned Namespace (ZNS) SSDs are the most recent evolution of host-managed flash-based storage, enabling improved performance at a lower cost-per-byte compared to traditional block interface SSDs. To date, there is no support for arranging these new devices in redundant arrays (RAID), which may limit their deployment in environments where this is the favored mechanism for increasing reliability and throughput. This paper identifies key challenges in the design of a RAID-like mechanism for ZNS SSDs, such as the requirement to manage metadata updates and persist partial stripe writes in the absence of overwrite semantics in the device’s interface. We present the design, implementation, and evaluation of RAIZN, a logical volume manager that exposes a ZNS interface and stripes data and parity across ZNS SSDs.

continued on page 17



Varuna problem setting and constraints.

PDL NEWS & AWARDS

July 2022

PDL Participates in CMU's Computer Science Scholars and AI4All Summer Programs

The CSS and AI4All @ Carnegie Mellon summer programs provide opportunities for students who have been historically excluded in STEM to study with full-time faculty, staff, and researchers who are leaders in the field. Students participated in project-based learning and attended lectures in various aspects of computing and artificial intelligence. They then applied those concepts and techniques to real world challenges and learn to leverage CS and AI for social good. Part of the 3-week residency for students attending these programs this year included a tour of the DCO with Jason Boles, Bill Courtright, Chuck Cranor, Mitch Franzos, and Greg Ganger.



June 2022

Brian Schwedock and Co-authors Nominated for Best Paper at ISCA '22!

Congratulations to Brian Schwedock and his co-authors on the nomination of "täkō: A Polymorphic Cache Hierarchy for General-Purpose Optimization of Data Movement" for Best Paper, presented at ISCA '22 in New York this June. täkō aims to improve software's ability to observe and respond to data movement, improving many inefficiencies, including the growing fraction of execution time and energy devoted to data movement itself.



May 2022

Introducing Sanay Rashmi Shah!

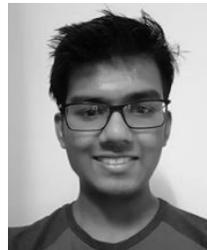
We would like to welcome the newest member of the PDL! Sanay Rashmi Shah was born on May 11, near midnight, weighing 5lb 15oz. His big sister Aanya is very excited and likes to read him bedtime stories. Parents Rashmi and Nihar are thrilled!



April 2022

Best Paper at EuroSys '22!

Congratulations to Sanjith Athlur, a new graduate student with the PDL, on achieving a Best Paper award at Eurosys 2022 for his work on "Varuna: Scalable, Low-cost Training of Massive Deep Learning Models." Varuna makes thrifty use of available networking resources and automatically configures the user's training job in massive deep learning models to efficiently use any given set of resources, reducing the effects of scalability limits on job parallelism and resource fragmentation. Sanjith will be working toward his Ph.D. in Computer Science with Rashmi Vinayak and Greg Ganger.



March 2022

1992 Paper Noted as Influential by ASPLOS '22

A 1992 paper written by Todd Mowry, PDL faculty and professor in the



Computer Science Department, and SCS alums Monica Lam and Anoop Gupta received an influential paper award at this year's International Conference on Architectural Support for Programming Languages (ASPLOS). The paper, "Design and Evaluation of a Compiler Algorithm for Prefetching", proposed a compiler algorithm to insert prefetch instructions into code that operates on dense matrices.

-- SCS WEDNESDAY WIRE: March 30, 2022

February 2022

Dimitrios Skarlatos and Collaborators Win Best Paper Award at ASPLOS '22!

Congratulations to Dimitrios Skarlatos, assistant professor in the Computer Science Department, for receiving a best paper award for "TMO: Transparent Memory Offloading in Datacenters" at ASPLOS 2022. ASPLOS is the premier forum for interdisciplinary systems research, intersecting computer architecture, hardware and emerging technologies, programming languages and compilers, operating systems and networking.

-- SCS WEDNESDAY WIRE: March 9, 2022



January 2022

Rashmi Vinayak Receives 2021 VMware Systems Research Award

Congratulations to Rashmi Vinayak, Assistant Professor of Electrical and Computer Engineering at CMU, on the occasion of her 2021 VMware Systems Research Award. The award



is given in recognition of the originality, impact and future potential of Rashmi's research, which focuses on work to enhance computer system reliability and efficiency by combining coding-theoretic algorithms, machine learning models and systems. See VMware's Award Announcement page for more details.

November 2021 PDL Paper wins ACM SoCC Test of Time Award!



The ACM SoCC 2021 test of time award goes to "Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis"! Congratulations to the authors Charles Reiss, Alexey Tumanov, Gregory R. Ganger, Randy H. Katz, and Michael A. Kozuch for their impactful work. The paper was first introduced at the 3rd ACM Symposium on Cloud Computing, held from October 14th-17th, 2012, in San Jose, CA. The paper helps its readers to better understand the challenges in developing effective cloudbased resource schedulers, analyzing the first publicly available trace data from a sizable multi-purpose cluster.

October 2021 Sara McAllister and Colleagues Win Best Paper at SOSP'21!

Congratulations to Sara McAllister, Benjamin Berg, Julian Tutuncu-Macias, Juncheng Yang, Sathya Gunasekar, Jimmy Lu, Daniel Berger, Nathan Beckmann, and Gregory R. Ganger



on receiving the Best Paper Award at SOSP'21, which was held virtually in Germany. The paper "Kangaroo: Caching Billions of Tiny Objects on Flash" introduces Kangaroo, a new flash-cache design that optimizes both DRAM usage and flash writes to maximize cache performance while minimizing cost.

October 2021 Mochi Project Wins 2021 R&D 100 Award

Mochi is a state-of-the-art open source tool for rapid development of customized data services supporting high-performance computing, big data, and large-scale learning across many scientific fields. Mochi allows developing these distributed services with a relatively small amount of new code while still enabling very high performance. Mochi was co-developed by researchers from Argonne National Laboratory, Carnegie Mellon University (George Amvrosiadis, Chuck Cranor, Qing Zheng), and Los Alamos National Laboratory. Established in 1963, the R&D 100 Awards is the only S&T (science and technology) awards competition that recognizes new commercial products, technologies and materials for their technological significance that are available for sale or license.



July 2021 PDL Researchers Win Best Paper at OSDI'21!

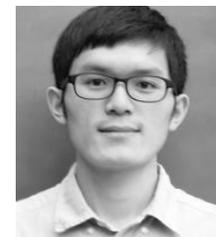
Congratulations to Aurick Qiao, Sang Keun Choe, Suhas Jayaram

Subramanya, Willie Neiswanger, Qirong Ho, Hao Zhang, Gregory R. Ganger, and Eric P. Xing, on being awarded one of three Best Paper Awards at OSDI'21. The paper, "Pollux: Co-adaptive Cluster Scheduling for Goodput-Optimized Deep Learning," discusses improved scheduling performance in deep learning (DL) clusters by adaptively co-optimizing inter-dependent factors both at the per-job level and at the cluster-wide level.



June 2021 Christos Faloutsos Promoted to University Professor

Congratulations to Christos, who has been elevated to the rank of University Professor, the highest distinction a faculty member can achieve at CMU.



June 2021 Huaicheng Li Recognized as Distinguished Reviewer

Congratulations to Huaicheng, who has been recognized as a Distinguished Reviewer for his service to System 2021, the 14th ACM International System and Storage Conference.

May 2021 CMU Research Forms Basis for Automatic Database Tuning Service

Dana Van Aken's dissertation has been transformed into a service aimed at improving the databases that power

continued on page 10

PDL NEWS & AWARDS

continued from page 9

popular websites. She founded the company with her advisor, Andy Pavlo, and Bohan Zhang, who worked with Van Aken and Pavlo as a research assistant.



OtterTune, a play on the once ubiquitous Auto-Tune, uses machine learning to automatically optimize databases, improving performance and efficiency and potentially saving companies time and money. Users could see faster loading times and improved services with a database humming along in the background. In case studies, the OtterTune improved efficiency

by 33% to 50%, cut one company's costs in half, and saved another tens of thousands of dollars.

"There are hundreds of settings to consider when optimizing a database, too many for humans to properly tune," said Dana Van Aken, "OtterTune takes human trial and error out of the mix." And from Pavlo: "Database management systems now exceed the administrator's ability to optimize them. We've put years of research into solving this problem, which we know will lead to significant increases in efficiencies and cost savings for customers."

-- info from CMU SCS News Wednesday, May 12, 2021

February 2021 Two Win Integrated Systems Apple Fellowship

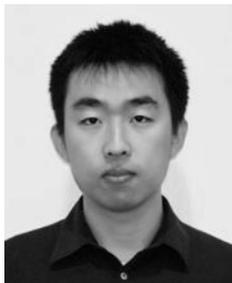
Congratulations to Minh S.Q. Truong and Mohammad Bakhshalipour, who have both received an Apple PhD Fellowship in Integrated Systems for the 2021-2023 academic years! The award includes a stipend and internships with Apple during the summers covered by the award.



NEW PDL FACULTY

Zhihao Jia

Assistant Professor, CSD



Welcome to the PDL! Zhihao is an assistant professor in the Computer Science Department at Carnegie Mellon University.

Before joining CMU, he was a research scientist at Facebook from 2020-2021. He received his PhD from the Computer Science Department at Stanford University in 2020, where he was co-advised by Alex Aiken and Matei Zaharia. Before Stanford, Zhihao received his bachelor's degree in Computer Science from the Special Pilot CS Class supervised by Andrew Yao at Tsinghua University.

Zhihao is interested in building systems for emerging application domains such as machine learning, quantum computing, and large-scale data analytics. In particular, his current research focuses on accelerating deep learning computations on modern hardware platforms, and optimiz-

ing quantum computations on today's intermediate-scale quantum devices. He is actively looking for strong and self-motivated students interested in building systems for machine learning and quantum computing to join his group.

Akshitha Sriraman

Assistant Professor, ECE

We would also like to welcome Akshitha Sriraman! She is an assistant professor in the Department of Electrical and Computer Engineering at Carnegie Mellon University. Her research interests are in the area of bridging computer architecture and software systems, with a focus on making hyperscale data centers more efficient (via solutions that span the systems stack). The central theme of her work is to design software that is aware of new hardware constraints/possibilities and architect hardware that efficiently supports new hyperscale software requirements. She has



recently been exploring ideas on designing "customized" hardware that can still generically support diverse applications, compiler-driven hardware design optimizations, novel software threading designs, and efficient I/O subsystems.

Akshitha's systems solutions to improve hardware efficiency have been deployed in real hyperscale data centers and currently serve billions of users, saving millions of dollars and meaningfully reducing the global carbon footprint. Additionally, her hardware design proposals have influenced the design of Intel's Alder Lake (Golden Cove and future generation) CPU architectures and its Infrastructure Processing Unit.

Akshitha's research has been recognized with an IEEE Micro Top Picks distinction and she was also awarded a Facebook Fellowship, a Rackham Merit Ph.D. Fellowship, and a CIS Full-Tuition Scholarship. She was selected to attend the Rising Stars in EECS Workshop and the Heidelberg Laureate Forum. She completed her Ph.D. in Computer Science and Engineering at the University of Michigan.

DISSERTATION ABSTRACT: A Principled Approach to Parallel Job Scheduling

Benjamin Samuel Berg
Carnegie Mellon University, SCS

PhD Defense — July 27, 2022

Parallelizable workloads are ubiquitous and appear across a diverse array of modern computer systems. Data centers, supercomputers, machine learning clusters, distributed computing frameworks, and databases all process jobs designed to be parallelized across many servers or cores. A job will receive some speedup from being parallelized across additional servers or cores, allowing the job to complete more quickly. However, jobs generally cannot be perfectly parallelized, and receive diminishing returns from being allocated additional cores. Hence, given a fixed number of cores, it is not obvious how to allocate cores across a set of jobs in order to reduce the response times of the jobs — the times from when each job arrives to the system until it is completed. While this question has been considered by the worst-case scheduling community, existing results are hampered by strong lower bounds on worst-case performance and tend to suggest policies which do not work well in practice. The goal of this thesis is to develop and analyze practical policies for scheduling parallelizable jobs using the tools of performance modeling and stochastic analysis.



PDL students and staff set up posters for the poster sessions at the 2022 PDL Spring Visit day.

Our approach in developing new scheduling policies for parallelizable jobs is threefold. First, we develop new stochastic models of parallelizable jobs running in a multicore system. Second, we analyze these new models using the tools of stochastic performance modeling, showing that a stochastic analysis emits scheduling policies which outperform the policies suggested by the worst-case literature. Finally, we validate our theoretical models through simulation and we develop new heuristic policies based on our theoretical results. We apply this methodology to a variety of practical settings where systems are required to schedule parallelizable jobs. We consider the case where job sizes are completely unknown to the system, the case where job sizes are perfectly known to the system, and the case where some jobs are known to be larger than others on average. Similarly, we consider the case where jobs are all assumed to have the same scaling behavior, the case where some jobs are more parallelizable than others, and the case where a job's parallelizability can change over time.

DISSERTATION ABSTRACT: Ultra-low-power, Energy- minimal Computer Architectures

Graham Gobieski
Carnegie Mellon University, SCS

PhD Defense — July 12, 2022

Ultra-low-power (ULP) sensor devices are increasingly being deployed for a variety of use-cases that require sophisticated processing of sensed data. Regardless of the deployment, energy-efficiency is critical; for battery-powered devices, energy-efficiency determines device lifetime, while for energy-harvesting devices, energy-efficiency determines performance by dictating the frequency of recharging. Unfortunately, existing devices pay a severe energy tax for their program-

mability, wasting energy in instruction-fetch/decode, pipeline-control and data supply. Further, offloading computation from an edge-device to the cloud is not practical as communication costs an order of magnitude more energy than local compute. The solution is to re-design the ULP sensor system stack to increase the energy-efficiency of on-board compute and enable sophisticated processing of sensed data. This thesis proposes such a stack — from software to silicon — that leverages new execution models to reduce the tax of programmability and achieve extreme energy-efficiency. Specifically it contributes 1) Sonic, a software framework that enables machine inference on intermittently-operating, energy-harvesting devices, 2) Manic, a vector-dataflow co-processor (and corresponding silicon prototype), 3) Snafu, an ULP coarse-grain-reconfigurable-array (CGRA) generation framework and architecture, and 4) RipTide, a co-designed dataflow compiler and energy-minimal CGRA. Sonic was the first demonstration of machine inference on an intermittent device, but also exposed the flaws of existing ULP devices. Manic fixed these problems by combining vector execution to amortize instruction fetch with dataflow execution to minimize data supply energy by forwarding intermediate values directly from producers to consumers. Snafu extended Manic's vector-dataflow to further reduce energy by minimizing the toggling of shared pipeline resources. Its generated CGRAs implement spatial-vector-dataflow execution that lays out computation across a fabric of PEs, keeping each PE configured in the same way throughout kernel execution. Finally, RipTide improves overall system efficiency by compiling and offloading to its CGRA, programs written in C with complex control-flow and irregular memory accesses. Together these contributions form the basis of a new ULP sensor system

continued on page 12

DEFENSES & PROPOSALS

continued from page 11

stack that is > 2 orders of magnitude more efficient than existing systems, enabling new emerging applications that require intelligence “beyond-the-edge.”

DISSERTATION ABSTRACT: Lightweight Preemptible Functions

Sol Boucher
Carnegie Mellon University, SCS

PhD Defense — February 24, 2022

We introduce novel programming abstractions for isolation of both time and memory. They operate at finer granularity than traditional primitives, supporting preemption at sub-millisecond timescales and tasks defined at the level of a function call. This resolution enables new functionality for application programmers, including users of unmanaged systems programming languages, all without requiring changes to the existing systems stack. Despite being concurrency abstractions, they employ synchronous invocation to allow application programmers to make their own scheduling decisions. However, we find that they compose naturally with existing concurrency abstractions centered around asynchronous background work, such as threads and futures. We demonstrate how such composition can enable asynchronous cancellation of threads and the implementation of preemptive thread libraries in userland, both regarded for decades as challenging problems.

DISSERTATION ABSTRACT: Towards Elastic and Resilient In-Network Computing

Daehyoek Kim
Carnegie Mellon University, SCS

PhD Defense — October 1, 2021

Recent advances in programmable networking hardware technology such as programmable switches and network interface cards create a new computing paradigm called in-network computing. This new paradigm allows functionality

that has been served by servers or proprietary hardware devices, ranging from network middleboxes to components of distributed systems, to now be performed in the network. The demand for higher performance and the commercial availability of programmable hardware have driven the popularity of in-network computing.

While many recent efforts have demonstrated the performance benefit of in-network computing, we observe that there is still a huge gap between what it offers today and evolving application demands. In particular, we argue that in-network computing lacks resource elasticity and fault resiliency which are essential building blocks for any practical computing platform, limiting its potential. Elasticity can address the shortcoming that today’s in-network computing only supports a simple deployment model where a single application runs on a single device equipped with fixed and limited resources. Similarly, fault resiliency is critical for managing prevalent device failures for the correctness and performance of applications, but it has gained little attention. Although resource elasticity and fault resiliency have been extensively studied for traditional server-based computing, we find that enabling them on programmable networking devices is challenging, especially due to their hardware constraints and workload characteristics.

In this thesis, we argue that by designing abstractions that effectively leverage resources available outside a single type of device while hiding the complexities of dealing with device heterogeneity, we can make in-network computing more elastic and resilient without any hardware modifications. This concept, which we call device resource augmentation, is a key enabler for resource elasticity and fault resiliency for stateful in-network applications written for programmable switches. In particular, we design three systems, named TEA, ExoPlane, and RedPlane, that use this concept to provide support for elastic memory and elastic compute/memory, and fault resiliency, respectively. Each of these systems consists of a key ab-

straction, programming APIs, and a runtime environment. We demonstrate their feasibility and effectiveness with prototype implementations and evaluations using various in-network applications.

DISSERTATION ABSTRACT: Self-Driving Database Management Systems: Forecasting, Modeling, and Planning

Lin Ma
Carnegie Mellon University, SCS

PhD Defense — August 18, 2021

Database management systems (DBMSs) are an important part of modern data-driven applications. However, they are notoriously difficult to deploy and administer because they have many aspects that one can change that affect their performance, including database physical design and system configuration. There are existing methods that recommend how to change these aspects of databases for an application. But most of them require humans to make final decisions on what changes to apply and when to apply them. Furthermore, these previous tuning methods either (1) require expensive exploratory testing, (2) are reactionary to the workload and can only solve problems after they occur, (3) focus only on improving one single aspect of the DBMS, or (4) do not provide explanations on their decisions. Thus, most DBMSs today still require onerous and costly human administration.

In this thesis, we present a novel architecture for a self-driving DBMS that enables automatic system management and removes the administration impediments. Our approach consists of three frameworks: (1) workload forecasting, (2) behavior modeling, and (3) action planning. The workload forecasting framework predicts the query arrival rates under varying database workload patterns using an ensemble of time-series forecasting models. The behavior modeling framework constructs fine-grained

continued on page 13

continued from page 12

machine learning models that predict the runtime behavior of the DBMS. Lastly, the action planning framework generates a sequence of optimization actions based on these forecasted workload patterns and behavior model estimations. It uses receding horizon control and Monte Carlo tree search to approximate the complex optimization problem effectively.

Our forecasting-modeling-planning architecture enables an autonomous DBMS that proactively plans for optimization actions without expensive testing. It automatically applies the actions at proper times, holistically controls all system aspects, and provides explanations on its decisions.

DISSERTATION ABSTRACT: Elastic Machine Learning Systems with Co-adaptation

Aurick Qiao
Carnegie Mellon University, SCS

PhD Defense — August 4, 2021

In recent years, the amount of computation being invested into machine learning (ML) and deep learning (DL) training has multiplied by several orders of magnitude. Under these conditions, elasticity (the ability of a system to dynamically adapt to changing supply and demand of compute resources over time) is a key ingredient for efficient resource management. Elasticity has long been proven to improve the resource utilization, execution performance, and fault tolerance of traditional applications such as web services and big data processing. However, elastic ML training is a relatively new area of interest, and faces different challenges from traditional applications due to ML training's highly sub-linear resource scalability, diverse execution patterns and strategies, and dependence between distributed workers.

This thesis steps beyond the existing early work in elastic ML by employing co-adaptation, i.e. combining both system-level and application-side adaptations, to better adapt to dynamic compute resources. Although previous frameworks

can enable elasticity by relying on system-level implementations, they ignore the inherent resource adaptability of ML training that can be leveraged to better overcome the aforementioned challenges. We present the design, implementation, and evaluation of three elastic systems for ML that improve DL training time in shared GPU clusters by 37-50%, enable elasticity for a diverse set of ML training applications, and reduce the impact of resource failures by 78-95%.

DISSERTATION ABSTRACT: Mitigating Memory-Safety Bugs with Efficient Out-of- Process Integrity Checking

Daming Dominic Chen
Carnegie Mellon University, SCS

PhD Defense — June 17, 2021

Computer programs written in low-level languages with manual memory management, like C and C++, can contain unintentional memory safety bugs due to developer error. Examples of these bugs include spatial buffer overflows, as well as temporal use-after-frees and double frees, which can be leveraged by attackers to exploit programs by altering their runtime behavior. Indeed, statistics from both Google Chrome and Microsoft show that ~70% of all security vulnerabilities in their codebases involve memory safety bugs.

Past work has proposed various strategies to eagerly detect or lazily mitigate such bugs. Eager approaches detect memory safety bugs by checking pointer operations, whereas lazy mitigations prevent exploitation by validating program data. To improve accuracy, mitigations may need to maintain internal state (metadata) about program execution, which must also be protected from corruption. This has been achieved using different techniques, including software-based address space partitioning, and hardware-based fine-grained instruction monitoring. Nevertheless, these approaches suffer from significant complexity, brittleness, or incompatibility, which reduces their efficiency and effectiveness.



Vikramraj Sitpal, Senior Member of Technical Staff, Oracle discusses Real World Challenges in the Oracle Database Cloud at the Spring 2022 Day of Consortium Talks.

In this thesis, we observe that existing mitigations are limited by their decision to maintain internal metadata within the same process. We show that augmenting hardware with a small, secure, and efficient AppendWrite inter-process communication (IPC) primitive allows metadata storage and policy checking to be performed in a separate isolated process, which improves both security and performance. We implement this design in our HerQules framework, which we show is capable of protecting both control-flow integrity and data-flow integrity. We evaluate our approach on a variety of real-world programs, including multiple benchmark suites, the NGINX web server, and the Google Chromium web browser.

DISSERTATION ABSTRACT: Modernizing Models and Management of the Memory Hierarchy for Non-Volatile Memory

Charles John McGuffey
Carnegie Mellon University, SCS

PhD Defense — May 3, 2021

Non-volatile memory technologies (NVMs) are a new family of technologies that combine near memory level performance with near storage level cost density. The result is a new type of memory hierarchy layer that exists and performs

continued on page 14

DEFENSES & PROPOSALS

continued from page 13

somewhere between the two. These new technologies offer many opportunities for performance improvement, but in order to take advantage of these system design needs to account for their adapted for their particular characteristics.

In this thesis, we focus on how to design memory management and caching systems for NVMs. Our work is broken into three major categories targeting different primary performance metrics. 1. We study how to design algorithms and memory management to achieve fault tolerance with low cost and efficient recovery using NVMs. 2. We design an extension the traditional model of caching to account for data writes in order to improve NVM device lifetime and energy consumption. 3. We investigate how to improve throughput in caches by taking advantage of granularity change in the memory hierarchy.

Throughout our work we rely on a blend of theoretical and practical approaches. We provide models for processor faults, cache writebacks, cache-storage communication, and trace complexity that isolate the targeted effects from orthogonal complications. For each model, we show worst case theoretical bounds for our algorithms along with proofs that explain how the benefits are derived. We then take our results and provide empirical evaluations to show their effectiveness in practice. We believe that our ideas and approach provide a solid foundational study on memory hierarchy design in the era of non-volatile memories.

DISSERTATION ABSTRACT: Dynamic Model Specialization for Efficient Inference, Training and Supervision

Ravi Teja Mullanpudi
Carnegie Mellon University, SCS

PhD Defense — April 26, 2021

With the advent of deep networks a significant focus in computer vision has been on building bigger and more accurate models for a variety of tasks using large amounts of human labeled data. This large scale supervised learning approach has well

known scalability challenges namely: 1) accurate general models are computationally expensive for training and inference 2) collecting and labeling large datasets requires extensive human effort and 3) datasets need to be repeatedly curated due to shifts in the target distribution. However, in many contexts, the goal is to build a model for a small fraction of the general distribution of images. In this thesis, rather than trying to build general models that are accurate in a wide range of contexts, we embrace the context specific view and present techniques for rapidly building and using models specialized to a context. Specifically, we exploit temporal specialization for building efficient video segmentation models. We show that continuously specializing a compact model to the content in a video stream enables accurate and efficient inference. We leverage specialization to visually similar categories for building efficient image classification architectures. We show that by specializing model features to discriminate between visually similar categories, one can improve inference efficiency by only computing the subset of features necessary for classifying a specific image. We explore specialization to individual categories for reducing human labeling effort in building models for rare categories. We show that models specialized to individual categories reduce human effort in mining large unlabeled data collections for examples relevant to rare categories. More broadly, we demonstrate that by dynamically specializing to a moment in time, to an input scene, or to a specific object category, it is possible to train accurate models quickly, reduce



PDL Director, Greg Ganger, opens the show at the 2022 Spring Visit Day, our first in person gathering since the Fall 2019 Retreat.

inference costs, and significantly reduce the amount of supervision required for training.

THESIS PROPOSAL: Beyond Model Efficiency: Data Optimizations for Machine Learning Systems

Michael Kuchnik, CSD
May 6, 2022

The field of machine learning has exploded due to the increased availability of data, compute, and algorithms. Systems built to support machine learning models have primarily focused on the compute path of the model itself. This thesis proposes to investigate the role of the data-path in both training and validation. For the first part of the thesis, we focus on training data, illustrating that the training data pipeline is a prime target for performance considerations. To aid in addressing performance issues, we introduce a form of training-pipeline subsampling, a reduced fidelity disk format, and a system for automatically tuning data pipeline performance knobs. In the second part, we propose to turn to the validation set of training, developing a system for automatically querying and validating a language model's behavior. We conclude with thoughts on how machine learning systems can expose data-friendly interfaces in upcoming generations of systems.

THESIS PROPOSAL: Leveraging Emerging Storage Technologies to Improve Performance, Reliability, and Availability

Thomas Sheek Kim, CSD
May 19, 2022

Zoned Namespace (ZNS) SSDs are the latest evolution of host-managed flash storage, enabling improved performance at a lower cost-per-byte than traditional block interface SSDs. However, the ZNS interface is incompatible with existing

continued on page 15

continued from page 14



Rekha Pitchumani, Sr. Research Manager, Samsung tells CMU personnel and industry visitors about Data-Centric Computing with Emerging Memory and Storage Devices at the Spring 2022 Day of Consortium Talks.

software, and introduces new challenges when deploying arrays of these SSDs.

This thesis builds on our existing work designing, implementing, and evaluating RAIZN, a system that exposes a logical volume with a zoned interface that provides RAID-like reliability and striping over ZNS SSDs. In this proposal, I detail our plans to implement and evaluate two major extensions to our existing system: Zone append support and configurable logical zone size.

THESIS PROPOSAL: Large-scale Machine Learning over Streaming Data

Ellango Jothimurugesan, CSD
March 29, 2022

Training data is constantly growing over time. This thesis investigates how to continuously update machine learning models in real-time both efficiently and with high accuracy, and with a focus on the statistical challenge of adapting to changes in the data distribution over time, known as concept drift. The proposed thesis will contribute several algorithms: (i) an optimization algorithm based on variance-reduced SGD that incorporates incremental data arrivals under time-varying arrival and processing rates for IID data; (ii) an adaptive algorithm for non-IID data arrivals that extends traditional concept drift detection tests into a stable-state/reactive-state process, attaining higher statistical accuracy and robustness to

hyperparameter selection; and (iii) a decentralized algorithm for data that is both non-IID both in time and space across nodes in a federated learning setting and allows for accurate collaborative training despite drifts. These algorithms have provable guarantees and experimentally outperform the state-of-the-art.

THESIS PROPOSAL: Ultra-low-power, Energy- minimal Computer Architectures

Graham Gobieski, CSD
Dec 6, 2021

Ultra-low-power (ULP) sensor devices are increasingly being deployed for a variety of use-cases in many different environments. The applications are wide-ranging and growing in complexity, relying on sophisticated techniques like on-device machine inference and advanced digital signal processing. However, existing systems suffer fundamental inefficiencies in supplying instructions and data that demand solutions across the compute stack: from software that enables sophisticated workloads on ULP devices to new, energy-minimal computer architectures.

The objective of this work is to design a complete system stack that leverages new execution models to maximize energy-efficiency without sacrificing programmability. Specifically this thesis contributes 1) SONIC, a software-runtime system that enables DNN inference on intermittent, embedded devices, 2) MANIC, an ultra-low-power vector-dataflow co-processor, 3) SNAFU, a energy-minimal CGRA generation framework and architecture, and 4) MANIC silicon, a silicon prototype of the MANIC co-processor. Taken together these systems form the foundation of a new stack with state-of-the-art energy efficiency, making sophisticated workloads practical, and support for programmability, allowing for iteration, development of new algorithms, and quick deployment.

Finally, to round out the new system stack, this work will discuss the ongoing characterization of the MANIC silicon prototype and the development of a new dataflow compiler to target SNAFU-like CGRAs.

THESIS PROPOSAL: Reliable and Resource-Efficient Learning Systems via Coding- Theory-Inspired Approaches

Jack Kosaian, CSD
November 10, 2021

Neural networks (NNs) are deployed in many settings, ranging from web services to safety critical systems. This has led to the development of various learning systems for deploying NNs. Learning systems must maintain high predictive performance (e.g., accuracy) while also meeting application-level objectives, such as latency constraints. However, meeting such application-level objectives is made challenging by two often-conflicting requirements: (1) Learning systems must operate reliably despite running atop unreliable, failure-prone hardware. (2) Learning systems must use hardware efficiently.

Techniques used to improve reliability often oppose those used to improve resource efficiency. To balance these conflicting goals, many computer systems leverage coding-theoretic tools, such as error-correcting codes and erasure codes. These tools have enabled resource-efficient reliability in storage, communication, and high-performance computing systems.

This thesis explores the use of ideas inspired by coding theory to improve the reliability and resource efficiency of learning systems. We pursue this through three main thrusts:

(1) We show how properties unique to learning systems can be exploited to more-efficiently integrate traditional coding-theoretic tools into learning systems. As an example, we reduce the execution-time overhead of fault-

continued on page 16

DEFENSES & PROPOSALS

continued from page 15

tolerant, safety-critical NN inference on GPUs by up to 5.3x by exploiting trends in NN design and GPU hardware.

(2) We demonstrate that co-designing coding-theoretic tools with learning systems offers new opportunities to extend the reach of these tools beyond their prior limitations. Specifically, we enable resource-efficient fault tolerance in distributed prediction serving systems by using machine learning to overcome a key barrier in prior coding-theoretic tools.

(3) We identify and exploit opportunities for coding-theory-inspired ideas to be used to improve the normal-mode performance of learning systems, when reliability is not a concern. We show that the throughput and GPU utilization of specialized convolutional neural network (CNN) inference can be improved by up to 2.5x through operating over images combined in a coding-theory-inspired manner and making appropriate modifications to the CNN architecture.

This thesis demonstrates the promise of using coding-theory-inspired tools in learning systems and aims to usher adoption of these tools in learning systems, similar to how they have been used in storage and communication systems.

THESIS PROPOSAL: On Building a Multiversed Cache Hierarchy with Page Overlays

Ziqi Wang, CSD
June 8, 2021

On modern multi-core architectures, the cache hierarchy serves as both a fast storage for frequently accessed data, and a communication channel between processor cores. Recent advancements in software and hardware, however, have motivated many interesting use cases of the cache that are not handled very well by today's cache hierarchy. This thesis proposal investigates into versioning, one of the most generally observed paradigms in daily programming but largely neglected



in cache system designs. We observe that many common problems from a wide range of applications can be addressed with hardware support for managing logically related data (“versions”). Current cache hierarchy struggles with these problems and forces software designs to adopt sub-optimal solutions, which leaves much space for improvement.

This proposal presents a systematic solution for several real-world problems that fall into the versioning category. We base our design on Page Overlays, a virtual memory framework that enables fine-grained address remapping. As part of our preliminary work towards this grand goal, we present OverlayTM, a Hardware Transactional Memory (HTM) design running the multi-versioned, serializable concurrency control protocol with a hardware commit queue. We also present NVOTM to leverage versions on different levels of the hierarchy to perform background redo logging onto NVM at millisecond-scale frequency. We next present MBC-2D as an inter-block cache compression architecture operating on the versioned “2D address space”, which achieves higher compression ratio with simpler metadata management.

This proposal also seeks to extend our preliminary works in the following major directions. First, we are eager to explore malloc-less object allocation using fast incache duplication of versions without the overhead of a software allocator. Second, we propose to extend the 2D inter-block compression domain from the last-level cache to the main memory, such that blocks are also orga-

nized in 2D compressed fashion in the main memory for more logical storage and less bus bandwidth consumption. Lastly, we also plan to investigate into page table compression in a virtualized environment. This is of high value to containers and microservice platforms where the page tables, as well as the processes themselves, are short-lived.

THESIS PROPOSAL: Compiler Techniques to Optimize Communication for Fragmented Applications

Pratik Fegade, CSD
June 2, 2021

With the increasing difference between communication and computation latencies, the performance of modern application domains such as deep learning and web applications is often bottlenecked by data movement. Given the size and complexity of these applications, their individual components are often (i) designed, developed, optimized and managed independently and therefore (ii) often use a wide array of different source languages as well as other technologies. Such fragmentation of the application's logic means that compiler optimizations, traditionally shown to be highly effective at optimizing data movement, may no longer be as effective because no compiler has an end-to-end view of the entire application.

In this thesis, we develop compiler techniques to optimize communication costs by (i) explicitly breaking artificial compilation boundaries arising due to fragmentation, and (ii) specializing for the structure of the data involved. We focus on microservice-based web applications and deep learning computations that exhibit shape and control flow dynamism. Accordingly, we first discuss our work on Cortex, a compiler that performs end-to-end optimization of the control flow and tensor computations found in recursive deep learning models. We then move on to discuss a few proposed directions where we aim

continued on page 17

continued from page 16

to (i) develop techniques for performant handling of control flow dynamism in deep learning, (ii) expand the scope of computations handled by tensor compilers and (iii) develop automated caching techniques for microservice applications. In solving these problems, we propose solutions that cross compilation and development boundaries, allowing us to perform global optimizations to reduce communication costs.

THESIS PROPOSAL: A Principled Approach to Parallel Job Scheduling

Benjamin Berg, CSD
May 10, 2021

Parallelizable workloads are ubiquitous and appear across a diverse array of modern computer systems. Data centers, supercomputers, machine learning clusters, distributed computing frameworks, and databases all process jobs designed to be parallelized across many servers or cores. A job will receive some

speedup from being parallelized across additional servers or cores, allowing the job to complete more quickly. However, jobs generally cannot be perfectly parallelized, and receive diminishing returns from being allocated additional servers. Hence, given a fixed number of servers, it is not obvious how to allocate servers across a set of jobs in order to reduce the response times of the jobs — the times from when each job arrives to the system until it is completed. While this question has been considered by the worst-case scheduling community, existing results are hampered by strong lower bounds on worst-case performance and tend to suggest policies which do not work well in practice. The goal of this thesis is to develop and analyze practical policies for scheduling parallelizable jobs using the tools of performance modeling and stochastic analysis.

Our approach in developing new scheduling policies for parallelizable jobs is threefold. First, we develop new stochastic models of parallelizable

jobs running in a multi-server system. Second, we analyze these new models using the tools of stochastic performance modeling, showing that a stochastic analysis emits scheduling policies which outperform the policies suggested by the worst-case literature. Finally, we validate our theoretical models through both simulation and real-world implementation to show that the scheduling policies we derive work well in practice. We apply this methodology to a variety of practical settings where systems are required to schedule parallelizable jobs. We consider the case where job sizes are completely unknown to the system, the case where job sizes are perfectly known to the system, and the case where job sizes are known to follow general distributions. Similarly, we consider the case where jobs are all assumed to have the same scaling behavior, the case where some jobs are more parallelizable than others, and the case where a job's parallelizability can change over time.

RECENT PUBLICATIONS

continued from page 7

Experiments show that RAZN provides full expected performance from the aggregate device set, successfully addressing the key challenges from the ZNS interface. RAZN achieves throughput and latency comparable to the equivalent Linux software RAID implementation running on conventional SSDs that use the same hardware platform, and then RAZN exceeds its performance once device-level garbage collection inhibits the conventional SSDs. Importantly, RAZN retains ZNS's opportunities for increased application performance, allowing higher-level software (e.g., F2FS or RocksDB) to carefully control garbage collection. This allows, for example, RAZN to maintain consistent performance under scenarios where conventional SSD arrays experience up to 87.5% throughput drop due to device-level garbage collection.

Are You Sure You Want to Use MMAP in Your Database Management System?

Andrew Crotty, Viktor Leis,
Andrew Pavlo

12th Annual Conference on Innovative
Data Systems Research (CIDR '22).
January 9-12, 2022, Chaminade,
USA.

Memory-mapped (mmap) file I/O is an OS-provided feature that maps the contents of a file on secondary storage into a program's address space. The program then accesses pages via pointers as if the file resided entirely in memory. The OS transparently loads pages only when the program references them and automatically evicts pages if memory fills up. mmap's perceived ease of use has seduced database management system (DBMS) develop-

ers for decades as a viable alternative to implementing a buffer pool. There are, however, severe correctness and performance issues with mmap that are not immediately apparent. Such problems make it difficult, if not impossible, to use mmap correctly and efficiently in a modern DBMS. In fact, several popular DBMSs initially used mmap to support larger-than-memory databases but soon encountered these hidden perils, forcing them to switch to managing file I/O themselves after significant engineering costs. In this way, mmap and DBMSs are like coffee and spicy food: an unfortunate combination that becomes obvious after the fact.

Since developers keep trying to use mmap in new DBMSs, we wrote this paper to provide a warning to others

continued on page 18

RECENT PUBLICATIONS

continued from page 17

that mmap is not a suitable replacement for a traditional buffer pool. We discuss the main shortcomings of mmap in detail, and our experimental analysis demonstrates clear performance limitations. Based on these findings, we conclude with a prescription for when DBMS developers might consider using mmap for file I/O.

FedLite: A Scalable Approach for Federated Learning on Resource-constrained Clients

Jianyu Wang, Hang Qi, Ankit Singh Rawat, Sashank Reddi, Sagar Waghmare, Felix X. Yu, Gauri Joshi

arXiv:2201.11865v2 [cs.LG] 16 February 2022.

In classical federated learning, the clients contribute to the overall training by communicating local updates for the underlying model on their private data to a coordinating server. However, updating and communicating the entire model becomes prohibitively expensive when resource-constrained clients collectively aim to train a large machine learning model. Split learning provides a natural solution in such a setting, where only a small part of the model is stored and trained on clients while the remaining large part of the model only stays at the servers.

However, the model partitioning employed in split learning introduces a significant amount of communication cost. This paper addresses this issue by compressing the additional communication using a novel clustering scheme accompanied by a gradient correction method. Extensive empirical evaluations on image and text benchmarks show that the proposed method can achieve up to $490\times$ communication cost reduction with minimal drop in accuracy, and enables a desirable performance vs. communication trade-off.

Leveraging Spatial and Temporal Correlations in Sparsified Mean Estimation

Divyansh Jhunjhunwala, Ankur Mallick, Advait Gadhikar, Swanand Kadhe, Gauri Joshi

35th Conference on Neural Information Processing Systems (NeurIPS 2021). Virtual Conference.

We study the problem of estimating at a central server the mean of a set of vectors distributed across several nodes (one vector per node). When the vectors are high dimensional, the communication cost of sending entire vectors may be prohibitive, and it may be imperative for them to use sparsifi-

cation techniques. While most existing work on sparsified mean estimation is agnostic to the characteristics of the data vectors, in many practical applications such as federated learning, there may be spatial correlations (similarities in the vectors sent by different nodes) or temporal correlations (similarities in the data sent by a single node over different iterations of the algorithm) in the data vectors. We leverage these correlations by simply modifying the decoding method used by the server to estimate the mean. We provide an analysis of the resulting estimation error as well as experiments for PCA, K-Means and Logistic Regression, which show that our estimators consistently outperform more sophisticated and expensive sparsification methods.

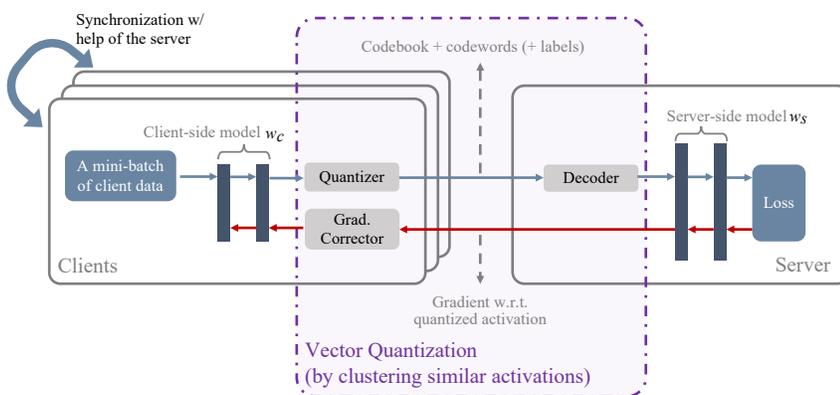
Arithmetic-Intensity-Guided Fault Tolerance for Neural Network Inference on GPUs

Jack Kosaian, K. V. Rashmi

SC'21, November 14–19, 2021, St. Louis, MO, USA.

Neural networks (NNs) are increasingly employed in safety-critical domains and in environments prone to unreliability (e.g., soft errors), such as on spacecraft. Therefore, it is critical to impart fault tolerance to NN inference. Algorithm-based fault tolerance (ABFT) is emerging as an efficient approach for fault tolerance in NNs.

We propose an adaptive approach to ABFT for NN inference that exploits untapped opportunities in emerging deployment scenarios. GPUs have high compute-to-memory-bandwidth ratios, while NN layers have a wide range of arithmetic intensities. This leaves some layers compute bound and others memory-bandwidth bound, but current approaches to ABFT do not consider these differences. We first investigate ABFT schemes best suited for each of these scenarios. We then propose intensity-guided ABFT, an adaptive, arithmetic-intensity-



Overview of the proposed algorithm. In order to reduce the additional communication associated with split learning, we propose to cluster similar client activations and only send the cluster centroids to the server. This is equivalent to adding a vector quantization layer in split neural network training. The success of our method relies on a novel variant of product quantizer and a gradient correction technique.

continued on page 19

continued from page 18

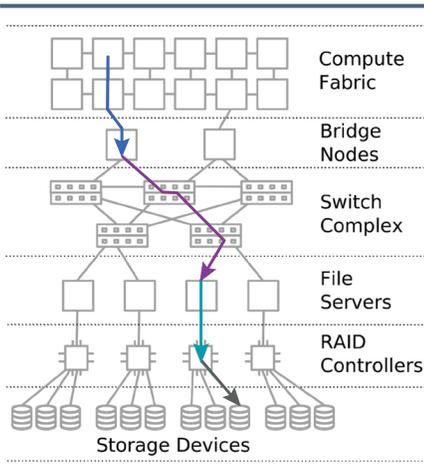
guided approach that selects the most efficient ABFT scheme for each NN layer. Intensity-guided ABFT reduces execution-time overhead by 1.09–5.3× across many NNs compared to traditional approaches to ABFT.

It's Time to Talk About HPC Storage: Perspectives on the Past and Future

Bradley Settlemyer, George Amvrosiadis, Philip Carns, Robert Ross

IEEE Computer Society Computing in Science & Engineering November/December 2021.

High-performance computing (HPC) storage systems are a key component of the success of HPC to date. Recently, we have seen major developments in storage related technologies, as well as changes to how HPC platforms are



Exemplar disaggregated HPC storage architecture. Traditional HPC storage systems have been propelled by simulation workloads to optimize for aggregate bulk synchronous throughput. This is a key disconnect for data-driven analysis: systems designed to maximize aggregate throughput are poorly suited to individual random reads. Each access must traverse multiple distinct protocol hops, where each protocol hop has its own interrupt processing, buffering, handshaking, serialization, and access control conventions. These protocol translations were designed in an era when high-latency storage devices gated overall performance, an assumption that no longer holds today.

used, especially in relation to artificial intelligence and experimental data analysis workloads. These developments merit a revisit of HPC storage system architectural designs. In this article, we discuss the drivers, identify key challenges to status quo posed by these developments, and discuss directions future research might take to unlock the potential of new technologies for the breadth of HPC applications.

The CoRa Tensor Compiler: Compilation for Ragged Tensors with Minimal Padding

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

arXiv:2110.10221v2 [cs.LG] 29 Oct 2021.

There is often variation in the shape and size of input data used for deep learning. In many cases, such data can be represented using tensors with non-uniform shapes, or ragged tensors. Due to limited and non-portable support for efficient execution on ragged tensors, current deep learning frameworks generally use techniques such as padding and masking to make the data shapes uniform and then offload the computations to optimized kernels for dense tensor algebra. Such techniques can, however, lead to a lot of wasted computation and therefore, a loss in performance. This paper presents CORA, a tensor compiler that allows users to easily generate efficient code for ragged tensor operators targeting a wide range of CPUs and GPUs. Evaluating CORA on a variety of operators on ragged tensors as well as on an encoder layer of the transformer model, we find that CORA (i) performs competitively with hand-optimized implementations of the operators and the transformer encoder and (ii) achieves, over PyTorch, a 1:6x geomean speedup for the encoder on an Nvidia GPU and a 1:86x geomean speedup for the multi-head attention module used in transformers on an ARM CPU.

IODA: A Host/Device Co-Design for Strong Predictability Contract on Modern Flash Storage

Huaicheng Li, Martin L. Putra, Ronald Shi, Xing Lin, Gregory R. Ganger, Haryadi S. Gunawi

SOSP '21, October 26–29, 2021, Virtual Event, Germany.

Predictable latency on flash storage is a long-pursuit goal, yet, unpredictability stays due to the unavoidable disturbance from many well-known SSD internal activities. To combat this issue, the recent NVMe IO Determinism (IOD) interface advocates host-level controls to SSD internal management tasks. While promising, challenges remain on how to exploit it for truly predictable performance.

We present IODA, an I/O deterministic flash array design built on top of small but powerful extensions to the IOD interface for easy deployment. IODA exploits data redundancy in the context of IOD for a strong latency predictability contract. In IODA, SSDs are expected to quickly fail an I/O on purpose to allow predictable I/Os through proactive data reconstruction. In the case of concurrent internal operations, IODA introduces busy remaining time exposure and predictable latency-window formulation to guarantee predictable data reconstructions. Overall, IODA only adds 5 new fields to the NVMe interface and a small modification in the flash firmware, while keeping most of the complexity in the host OS. Our evaluation shows that IODA improves the 95±99.99th latencies by up to 75×. IODA is also the nearest to the ideal, no disturbance case compared to 7 state-of-the-art preemption, suspension, GC coordination, partitioning, tiny-tail flash controller, prediction, and proactive approaches.

continued on page 20

RECENT PUBLICATIONS

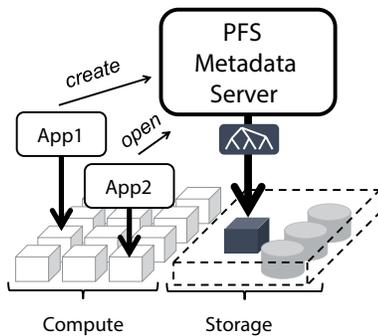
continued from page 19

DeltaFS: A Scalable No-Ground-Truth Filesystem For Massively-Parallel Computing

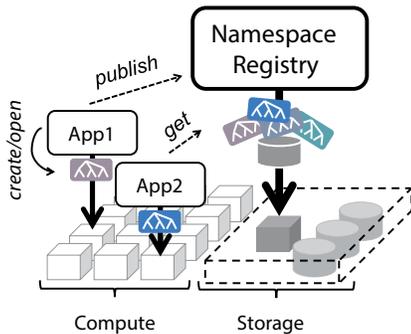
Qing Zheng, Chuck Cranor, Greg Ganger, Garth Gibson, George Amvrosiadis, Brad Settlemyer, Gary Grider

SC '21, November 14–19, 2021, St. Louis, MO, USA.

High-Performance Computing (HPC) is known for its use of massive concurrency. But it can be challenging for a parallel filesystem's control plane to utilize cores when every client process must globally synchronize and serialize its metadata mutations with those of other clients. We present DeltaFS, a new paradigm for distributed filesystem metadata.



(a) Today's Parallel Filesystems



(b) Our Vision: DeltaFS

Motivation for no ground truth. Rather than synchronously communicating with a slow global filesystem namespace, jobs communicate instead by publishing and sharing filesystem namespace snapshots on an as-needed basis through a public registry for high performance.

DeltaFS allows jobs to self-commit their namespace changes to logs, avoiding the cost of global synchronization. Followup jobs selectively merge logs produced by previous jobs as needed, a principle we term No Ground Truth which allows for efficient data sharing. By avoiding unnecessary synchronization of metadata operations, DeltaFS improves metadata operation throughput up to 98× leveraging parallelism on the nodes where job processes run. This speedup grows as job size increases. DeltaFS enables efficient inter-job communication, reducing overall workflow runtime by significantly improving client metadata operation latency up to 49× and resource usage up to 52×.

The Case for Phase-Aware Scheduling of Parallelizable Jobs

Ben Berg, Justin Whitehouse, Ben Moseley, Weina Wang, Mor Harchol-Balter

IFIP Performance 2021. Milan, Italy, November 2021.

Parallelizable jobs typically consist of multiple phases of computation, where the job is more parallelizable in some phases and less parallelizable in others. For example, in a database, a query may consist of a highly parallelizable table scan, followed by a less parallelizable table join. In the past, this phase-varying parallelizability was summarized by a single sub-linear speedup curve that measures a job's average parallelizability over its entire lifetime. Today, however, modern systems have fine-grained knowledge of the exact phase each job is in at every moment in time. Unfortunately, these systems do not fully leverage this real-time feedback when scheduling parallelizable jobs. Theory has failed to produce practical phase-aware scheduling policies, and thus scheduling in current systems is largely heuristic.

A phase-aware scheduling policy must decide, given its knowledge of each job's current phase, how many servers

or cores to allocate to each job in the system at every moment in time. This paper provides the first stochastic model of a system processing parallelizable jobs composed of phases. Using our model, we derive an optimal phase-aware scheduling policy that minimizes the mean response time across jobs. Our provably optimal policy, Inelastic-First (IF), gives strict priority to jobs that are currently in less parallelizable phases. We validate our theoretical results using a simulation of a database running queries from the Star Schema Benchmark. We compare IF to a range of policies from both systems and theory, and show that IF reduces mean response time by up to a factor of 3.

The Most Common Queueing Theory Questions Asked by Computer Systems Practitioners

Mor Harchol-Balter and Ziv Scully

First International Workshop on Teaching Performance Analysis of Computer Systems (TeaPACS 2021) In conjunction with the IFIP Performance 2021 Conference. Milan, Italy, Nov. 2021.

This document examines five performance questions which are repeatedly asked by practitioners in industry: (i) My system utilization is very low, so why are job delays so high? (ii) What should I do to lower job delays? (iii) How can I favor short jobs if I don't know which jobs are short? (iv) If some jobs are more important than others, how do I negotiate importance versus size? (v) How do answers change when dealing with a closed-loop system, rather than an open system? All these questions have simple answers through queueing theory. This short paper elaborates on the questions and their answers. To keep things readable, our tone is purposely informal throughout.

continued on page 21

continued from page 20

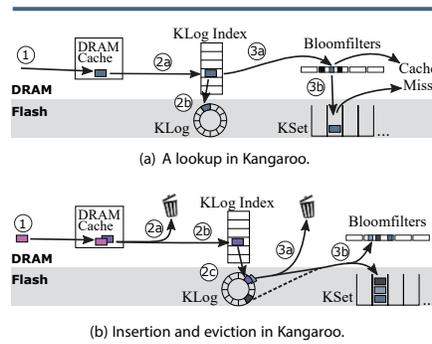
Kangaroo: Caching Billions of Tiny Objects on Flash

Sara McAllister, Benjamin Berg, Julian Tutuncu-Macias, Juncheng Yang, Sathya Gunasekar, Jimmy Lu, Daniel Berger, Nathan Beckmann, Gregory R. Ganger

28th ACM Symposium on Operating Systems Principles (SOSP '21) October 25-28, 2021. Virtual Event. BEST PAPER!

Many social-media and IoT services have very large working sets consisting of billions of tiny (~100 B) objects. Large, flash-based caches are important to serving these working sets at acceptable monetary cost. However, caching tiny objects on flash is challenging for two reasons: (i) SSDs can read/write data only in multi-KB pages that are much larger than a single object, stressing the limited number of times flash can be written; and (ii) very few bits per cached object can be kept in DRAM without losing flash's cost advantage. Unfortunately, existing flash-cache designs fall short of addressing these challenges: write-optimized designs require too much DRAM, and DRAM-optimized designs write flash too much.

We present Kangaroo, a new flash-cache design that optimizes both DRAM usage and flash writes to maximize cache performance while minimizing cost. Kangaroo combines a large, set-associative cache with a small, log-structured cache. The set-associative cache requires minimal DRAM, while the log-structured cache minimizes Kangaroo's flash writes. Experiments using traces from Facebook and Twitter show that Kangaroo achieves DRAM usage close to the best prior DRAM-optimized design, flash writes close to the best prior write-optimized design, and miss ratios better than both. Kangaroo's design is Pareto-optimal across a range of allowed write rates, DRAM sizes, and flash sizes, reducing misses by 29% over the state of the art. These results



Overview of Kangaroo. Objects first go to a tiny DRAM cache; then KLog, a small on-flash log-structured cache with an in-DRAM index; and finally KSet, a large on-flash set-associative cache. KLog minimizes flash writes, and KSet minimizes DRAM usage.

are corroborated with a test deployment of Kangaroo in a production flash cache at Facebook.

The Gittins Policy in the M/G/1 Queue

Ziv Scully, Mor Harchol-Balter

19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt 2021) Philadelphia, PA, Oct 2021.

The Gittins policy is a highly general scheduling policy that minimizes a wide variety of mean holding cost metrics in the M/G/I queue. Perhaps most famously, Gittins minimizes mean response time in the M/G/I when jobs' service times are unknown to the scheduler. Gittins also minimizes weighted versions of mean response time. For example, the well-known "cμ rule", which minimizes class-weighted mean response time in the multiclass M/M/I, is a special case of Gittins. However, despite the extensive literature on Gittins in the M/G/I, it contains no fully general proof of Gittins's optimality. This is because Gittins was originally developed for the multiarmed bandit problem. Translating arguments from the multiarmed bandit to the M/G/I is technically demanding, so it has only been done rigorously in some special cases. The

extent of Gittins's optimality in the M/G/I is thus not entirely clear. In this work we provide the first fully general proof of Gittins's optimality in the M/G/I. The optimality result we obtain is even more general than was previously known. For example, we show that Gittins minimizes mean slowdown in the M/G/I with unknown or partially known service times, and we show that Gittins's optimality holds under batch arrivals. Our proof uses a novel approach that works directly with the M/G/I, avoiding the difficulties of translating from the multi-armed bandit problem.

Personalized Federated Learning for Heterogeneous Clients with Clustered Knowledge Transfer

Yae Jee Cho, Jianyu Wang, Tarun Chiruvolu, Gauri Joshi

arXiv:2109.08119v1 [cs.LG] 16 Sep. 2021.

Personalized federated learning (FL) aims to train model(s) that can perform well for individual clients that are highly data and system heterogeneous. Most work in personalized FL, however, assumes using the same model architecture at all clients and increases the communication cost by sending/receiving models. This may not be feasible for realistic scenarios of FL. In practice, clients have highly heterogeneous system-capabilities and limited communication resources. In our work, we propose a personalized FL framework, PerFed-CKT, where clients can use heterogeneous model architectures and do not directly communicate their model parameters. PerFed-CKT uses clustered co-distillation, where clients use logits to transfer their knowledge to other clients that have similar data-distributions. We theoretically show the convergence and generalization properties of PerFed-CKT and empirically show that PerFed-CKT achieves high test accuracy with several orders

continued on page 22

RECENT PUBLICATIONS

continued from page 21

of magnitude lower communication cost compared to the state-of-the-art personalized FL schemes.

Rateless Codes for Distributed Non-linear Computations

Ankur Mallick, Sophie Smith, Gauri Joshi

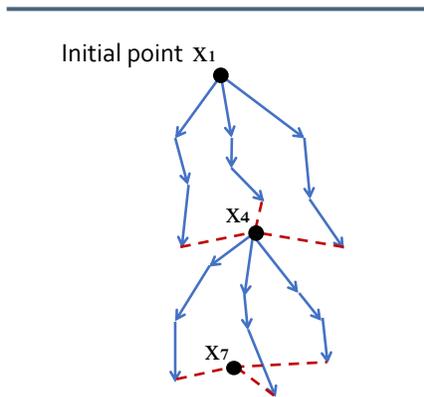
International Symposium on Topics in Coding, Sept 2021.

Machine learning today involves massive distributed computations running on cloud servers, which are highly susceptible to slowdown or straggling. Recent work has demonstrated the effectiveness of erasure codes in mitigating such slowdown for linear computations, by adding redundant computations such that the entire computation can be recovered as long as a subset of nodes finish their assigned tasks. However, most machine learning algorithms typically involve non-linear computations that cannot be directly handled by these coded computing approaches. In this work, we propose a coded computing strategy for mitigating the effect of stragglers on non-linear distributed computations. Our strategy relies on the observation that many expensive non-linear functions can be decomposed into sums of cheap non-linear functions. We show that erasure codes, specifically rateless codes can be used to generate and compute random linear combinations of these functions at the nodes such that the original function can be computed as long as a subset of nodes return their computations. Simulations and experiments on AWS Lambda demonstrate the superiority of our approach over various uncoded baselines.

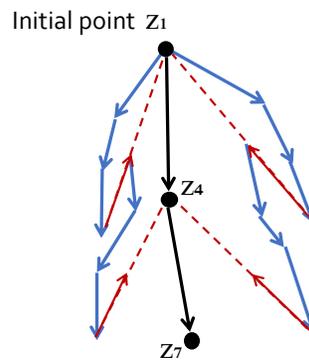
Cooperative SGD: A Unified Framework for the Analysis of Local-Update SGD

Jianguo Wang, Gauri Joshi

Journal of Machine Learning Research (JMLR), 2021.



(a) Periodic simple-averaging SGD.



(b) Elastic averaging SGD.

Illustration of PSASGD and EASGD in the model parameter space. Blue and black arrows denote local SGD iterations and the update of auxiliary variables, respectively. Red arrows represent averaging local models with each other or with the auxiliary variable. In this toy example, the number of local updates τ is set to 3.

When training machine learning models using stochastic gradient descent (SGD) with a large number of nodes or massive edge devices, the communication cost of synchronizing gradients at every iteration is a key bottleneck that limits the scalability of the system and hinders the benefit of parallel computation. Local-update SGD algorithms, where worker nodes perform local iterations of SGD and periodically synchronize their local models, can effectively reduce the communication frequency and save the communication delay. In this paper, we propose a powerful framework, named Cooperative SGD, that subsumes a variety of

local-update SGD algorithms (such as local SGD, elastic averaging SGD, and decentralized parallel SGD) and provides a unified convergence analysis. Notably, special cases of the unified convergence analysis provided by the cooperative SGD framework yield 1) the first convergence analysis of elastic averaging SGD for general non-convex objectives, and 2) improvements upon previous analyses of local SGD and decentralized parallel SGD. Moreover, we design new algorithms such as elastic averaging SGD with overlapped computation and communication, and decentralized periodic averaging which are shown to be 4x or more faster than the baseline in reaching the same training loss.

A Novel Framework for the Analysis and Design of Heterogeneous Federated Learning

Jianguo Wang, Qinghua Liu, Hao Liang, Gauri Joshi and H. Vincent Poor

IEEE Transactions on Signal Processing, Sept 2021.

In federated learning, heterogeneity in the clients' local datasets and computation speeds results in large variations in the number of local updates performed by each client in each communication round. Naive weighted aggregation of such models causes objective inconsistency, that is, the global model converges to a stationary point of a mismatched objective function which can be arbitrarily different from the true objective. This paper provides a general framework to analyze the convergence of federated optimization algorithms with heterogeneous local training progress at clients. The analyses are conducted for both smooth non-convex and strongly convex settings, and can also be extended to partial client participation case. Additionally, it subsumes previously proposed methods such as FedAvg and FedProx, and provides

continued on page 23

continued from page 22

the first principled understanding of the solution bias and the convergence slowdown due to objective inconsistency. Using insights from this analysis, we propose FedNova, a normalized averaging method that eliminates objective inconsistency while preserving fast error convergence.

ZNS: Avoiding the Block Interface Tax for Flash-based SSDs

Matias Bjørling, Abutalib Aghayev, Hans Holmberg, Aravind Ramesh, Damien Le Moal, Gregory R. Ganger, George Amvrosiadis

USENIX Annual Technical Conference (USENIX 2021), July 14-16, 2021, Virtual Event.

The Zoned Namespace (ZNS) interface represents a new division of functionality between host software and flash-based SSDs. Current flash-based SSDs maintain the decades-old block interface, which comes at substantial expense in terms of capacity over-provisioning, DRAM for page mapping tables, garbage collection overheads, and host software complexity attempting to mitigate garbage collection. ZNS offers shelter from this ever-rising block interface tax.

This paper describes the ZNS interface and explains how it affects both SSD hardware/firmware and host software. By exposing flash erase block boundaries and write-ordering rules, the ZNS interface requires the host software to address these issues while continuing to manage media reliability within the SSD. We describe how storage software can be specialized to the semantics of the ZNS interface, often resulting in significant efficiency benefits. We show the work required to enable support for ZNS SSDs, and show how modified versions of f2fs and RocksDB take advantage of a ZNS SSD to achieve higher throughput and lower tail latency as compared to running on a block-interface SSD with identical physical hardware. For example,

we find that the 99.9th-percentile random-read latency for our zone-specialized RocksDB is at least 2-4x lower on a ZNS SSD compared to a blockinterface SSD, and the write throughput is 2x higher.

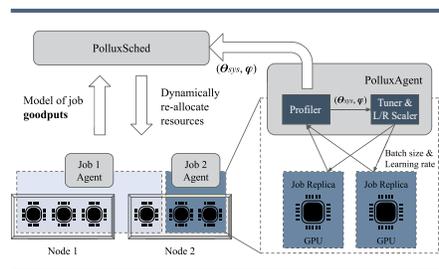
Pollux: Co-adaptive Cluster Scheduling for Goodput-Optimized Deep Learning

Aurick Qiao, Sang Keun Choe, Suhas Jayaram Subramanya, Willie Neiswanger, Qirong Ho, Hao Zhang, Gregory R. Ganger, Eric P. Xing

15th USENIX Symposium on Operating Systems Design and Implementation, Virtual Event, July 14-16, 2021. BEST PAPER!

Pollux improves scheduling performance in deep learning (DL) clusters by adaptively co-optimizing inter-dependent factors both at the per-job level and at the cluster-wide level. Most existing schedulers expect users to specify the number of resources for each job, often leading to inefficient resource use. Some recent schedulers choose job resources for users, but do so without awareness of how DL training can be re-optimized to better utilize the provided resources.

Pollux simultaneously considers both aspects. By monitoring the status of each job during training, Pollux models how their goodput (a metric we introduce to combine system throughput with statistical efficiency) would change by adding or removing resources. Pollux dynamically (re-)assigns resources to improve cluster-wide goodput,



The co-adaptive scheduling architecture of Pollux.

while respecting fairness and continually optimizing each DL job to better utilize those resources.

In experiments with real DL jobs and with trace-driven simulations, Pollux reduces average job completion times by 37-50% relative to state-of-the-art DL schedulers, even when they are provided with ideal resource and training configurations for every job. Pollux promotes fairness among DL jobs competing for resources, based on a more meaningful measure of useful job progress, and reveals a new opportunity for reducing DL cost in cloud environments. Pollux is implemented and publicly available as part of an open-source project at <https://github.com/petuum/adaptld>.

Progressive Compressed Records: Taking a Byte out of Deep Learning Data

Michael Kuchnik, George Amvrosiadis, Virginia Smith

Proceedings of the VLDB Endowment, Vol. 14, No. 11 ISSN 2150-8097, July 2021.

Deep learning accelerators efficiently train over vast and growing amounts of data, placing a newfound burden on commodity networks and storage devices. A common approach to conserve bandwidth involves resizing or compressing data prior to training. We introduce Progressive Compressed Records (PCRs), a data format that uses compression to reduce the overhead of fetching and transporting data, effectively reducing the training time required to achieve a target accuracy. PCRs deviate from previous storage formats by combining progressive compression with an efficient storage layout to view a single dataset at multiple fidelities—all without adding to the total dataset size. We implement PCRs and evaluate them on a range of datasets, training tasks, and hardware architectures. Our work shows that: (i)

continued on page 24

RECENT PUBLICATIONS

continued from page 23

the amount of compression a dataset can tolerate exceeds 50% of the original encoding for many DL training tasks; (ii) it is possible to automatically and efficiently select appropriate compression levels for a given task; and (iii) PCRs enable tasks to readily access compressed data at runtime—utilizing as little as half the training bandwidth and thus potentially doubling training speed.

Block-Granularity-Aware Caching

Nathan Beckmann, Phil Gibbons, Charles McGuffey

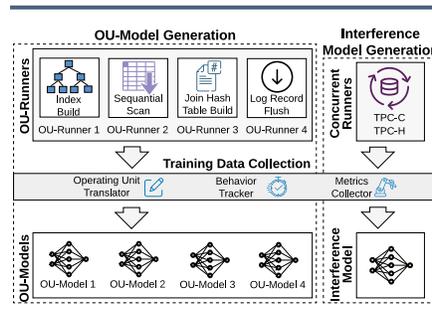
SPAA '21: Proceedings of the 33rd ACM Symposium on Parallelism in Algorithms and Architectures. July 2021. A common feature of computer systems is that block granularity changes at different levels of the storage hierarchy. This paper presents the first study of how granularity change affects caching. We define the Block-Granularity-Aware (BGA) Caching Model, prove new adversarial competitive bounds for the problem, and develop an on-line BGA caching policy with a better competitive ratio than traditional cache policies in this setting.

MB2: Decomposed Behavior Modeling for Self-Driving Database Management Systems

Lin Ma, William Zhang, Jie Jiao, Wuwen Wang, Matthew Butrovich, Wan Shen Lim, Prashanth Menon, Andrew Pavlo

Proceedings of SIGMOD, pages. 1248–1261, June 2021.

Database management systems (DBMSs) are notoriously difficult to deploy and administer. The goal of a self-driving DBMS is to remove these impediments by managing itself automatically. However, a critical problem in achieving full autonomy is how to predict the DBMS's runtime behavior



System Architecture – MB2 trains OU-models for each OU with the data generated by OU-runners and builds interference models for concurrent OUs based on system resource utilization.

and resource consumption. These predictions guide a self-driving DBMS's decision-making components to tune and optimize all aspects of the system. We present the ModelBot2 end-to-end framework for constructing and maintaining prediction models using machine learning (ML) in self-driving DBMSs. Our approach decomposes a DBMS's architecture into fine-grained operating units that make it easier to estimate the system's behavior for configurations that it has never seen before. ModelBot2 then provides an offline execution environment to exercise the system to produce the training data used to train its models. We integrated ModelBot2 in an in-memory DBMS and measured its ability to predict its performance for OLTP and OLAP workloads running in dynamic environments. We also compare ModelBot2 against state-of-the-art ML models and show that our models are up to 25× more accurate in multiple scenarios.

Boosting the Throughput and Accelerator Utilization of Specialized CNN Inference Beyond Increasing Batch Size

Jack Kosaian, Amar Phanishayee, Matthai Philipose, Debadeepta Dey, K. V. Rashmi

38th International Conference on Machine Learning, PMLR 139, 18–24 July 2021, Virtual Event.

Datacenter vision systems widely use small, specialized convolutional neural networks (CNNs) trained on specific tasks for high-throughput inference. These settings employ accelerators with massive computational capacity, but which specialized CNNs under-utilize due to having low arithmetic intensity. This results in suboptimal application-level throughput and poor returns on accelerator investment. Increasing batch size is the only known way to increase both application-level throughput and accelerator utilization for inference, but yields diminishing returns; specialized CNNs poorly utilize accelerators even with large batch size. We propose FoldedCNNs, a new approach to CNN design that increases inference throughput and utilization beyond large batch size. FoldedCNNs rethink the structure of inputs and layers of specialized CNNs to boost arithmetic intensity: in FoldedCNNs, f images with C channels each are concatenated into a single input with fC channels and jointly classified by a wider CNN. Increased arithmetic intensity in FoldedCNNs increases the throughput and GPU utilization of specialized CNN inference by up to 2.5x and 2.8x, with accuracy close to the original CNN in most cases.

Ripple: Profile-Guided Instruction Cache Replacement for Data Center Applications

Tanvir Ahmed Khan, Dexin Zhang, Akshitha Sriraman, Joseph Devietti, Gilles A Pokam, Heiner Litz, Baris Kasikci

International Symposium on Computer Architecture (ISCA), June 2021. Modern data center applications exhibit deep software stacks, resulting in large instruction footprints that frequently cause instruction cache misses degrading performance, cost, and energy efficiency. Although numerous mechanisms have been proposed to mitigate instruction cache misses, they

continued on page 25

continued from page 24

still fall short of ideal cache behavior, and furthermore, introduce significant hardware overheads. We first investigate why existing I-cache miss mitigation mechanisms achieve sub-optimal performance for data center applications. We find that widely-studied instruction prefetchers fall short due to wasteful prefetch-induced cache line evictions that are not handled by existing replacement policies. Existing replacement policies are unable to mitigate wasteful evictions since they lack complete knowledge of a data center application’s complex program behavior. To make existing replacement policies aware of these eviction-inducing program behaviors, we propose Ripple, a novel software-only technique that profiles programs and uses program context to inform the underlying replacement policy about efficient replacement decisions. Ripple carefully identifies program contexts that lead to I-cache misses and sparingly injects “cache line eviction” instructions in suitable program locations at link time. We evaluate Ripple using nine popular data center applications and demonstrate that Ripple enables any replacement policy to achieve speedup that is closer to that of an ideal I-cache. Specifically, Ripple achieves an average performance improvement of 1.6% (up to 2.13%) over prior work due to a mean 19% (up to 28.6%) I-cache miss reduction.

The Processing-in-Memory Model

H. Kang, P. B. Gibbons, G. E. Blelloch, L. Dhulipala, Y. Gu, C. McGuffey

SPAA ‘21: Proceedings of the 33rd ACM Symposium on Parallelism in Algorithms and Architectures. July 2021.

As computational resources become more efficient and data sizes grow, data movement is fast becoming the dominant cost in computing. Processing-in-Memory is emerging as a key technique for reducing costly data movement, by enabling computation to be executed on compute resources embedded in the memory modules themselves.

This paper presents the Processing-in-Memory (PIM) model, for the design and analysis of parallel algorithms on systems providing processing-in-memory modules. The PIM model focuses on key aspects of such systems, while abstracting the rest. Namely, the model combines (i) a CPU-side consisting of parallel cores with fast access to a small shared memory of size M words (as in traditional parallel computing), (ii) a PIM-side consisting of P PIM modules, each with a core and a local memory of size $\Theta(n/P)$ words for an input of size n (as in traditional distributed computing), and (iii) a network between the two sides.

The model combines standard parallel complexity metrics for both shared memory (work and depth) and distributed memory (local work, communication time) computing. A key algorithmic challenge is to achieve load balance among the PIM modules in both their communication and their local work, while minimizing the communication time. We demonstrate how

to overcome this challenge for an ordered search structure, presenting a parallel PIM-skiplist data structure that efficiently supports a wide range of batch-parallel queries and updates.

Spitfire: A Three-Tier Buffer Manager for Volatile and Non-Volatile Memory

Xinjing Zhou, Joy Arulraj, Andrew Pavlo, David Cohen

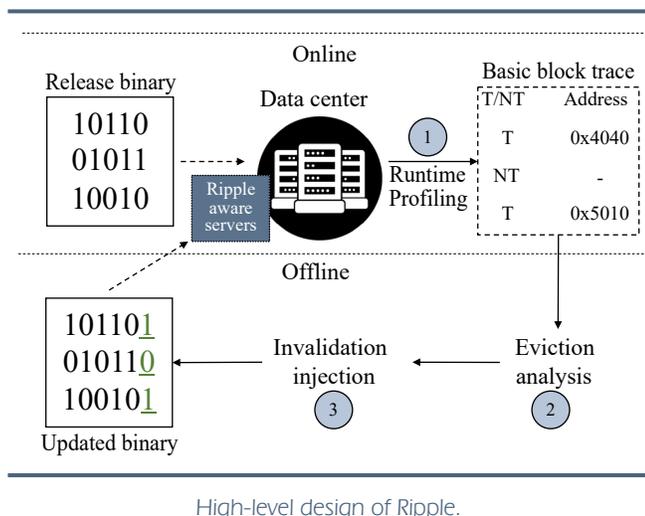
SIGMOD/PODS ‘21: Proceedings of the 2021 International Conference on Management of Data. June 2021.

The design of the buffer manager in database management systems (DBMSs) is influenced by the performance characteristics of volatile memory (i.e., DRAM) and non-volatile storage (e.g., SSD). The key design assumptions have been that the data must be migrated to DRAM for the DBMS to operate on it and that storage is orders of magnitude slower than DRAM. But the arrival of new non-volatile memory (NVM) technologies that are nearly as fast as DRAM invalidates these previous assumptions.

Researchers have recently designed Hymem, a novel buffer manager for a three-tier storage hierarchy comprising of DRAM, NVM, and SSD. Hymem supports cache-line-grained loading and an NVM-aware data migration policy. While these optimizations improve its throughput, Hymem suffers from two limitations. First, it is a single-threaded buffer manager. Second, it is evaluated on an NVM emulation platform. These limitations constrain the utility of the insights obtained using Hymem.

In this paper, we present Spitfire, a multi-threaded, three-tier buffer manager that is evaluated on real NVM hardware. We introduce a general framework for reasoning about data migration in a multi-tier storage hierarchy. We illustrate the limitations of the optimizations used in Hymem on

continued on page 26



High-level design of Ripple.

RECENT PUBLICATIONS

continued from page 25

Optane and then discuss how Spitfire circumvents them. We demonstrate that the data migration policy has to be tailored based on the characteristics of the devices and the workload. Given this, we present a machine learning technique for automatically adapting the policy for an arbitrary workload and storage hierarchy. Our experiments show that Spitfire works well across different workloads and storage hierarchies.

WineFS: A Hugepage-aware File System for Persistent Memory that Ages Gracefully

Rohan Kadekodi, Saurabh Kadekodi, Soujanya Ponnappalli, Harshad Shirwadkar, Gregory R. Ganger, Aasheesh Kolli, Vijay Chidambaram

28th ACM Symposium on Operating Systems Principles (SOSP '21) October 25-28, 2021.

Modern persistent-memory (PM) file systems perform well in benchmark settings, when the file system is freshly created and empty. But after being aged by usage, as will be the normal mode in practice, their memory-mapped performance degrades significantly. This paper shows that the cause is their

inability to use 2MB hugepages to map files when aged, having to use 4KB pages instead and suffering many extra page faults and TLB misses as a result.

We introduce WineFS, a novel hugepage-aware PM file system that largely eliminates this effect. WineFS combines a new alignment-aware allocator with fragmentation-avoiding approaches to consistency and concurrency to preserve the ability to use hugepages. Experiments show that WineFS resists the effects of aging and outperforms state-of-the-art PM file systems in both aged and un-aged settings. For example, in an aged setup, the LMDB memory-mapped database obtains 2× higher write throughput on WineFS compared to NOVA, and 70% higher throughput compared to ext4- DAX. When reading a memory-mapped persistent radix tree, WineFS results in 56% lower median latency than NOVA.

Filter Representation in Vectorized Query Execution

Amadou Ngom, Prashanth Menon, Matthew Butrovich, Lin Ma, Wan Shen Lim, Todd C. Mowry, Andrew Pavlo

International Workshop on Data Management on New Hardware, pp. 6:1–6:7, June 2021.

Advances in memory technology have made it feasible for database management systems (DBMS) to store their working data set in main memory. This trend shifts the bottleneck for query execution from disk accesses to CPU efficiency. One technique to improve CPU efficiency is batch-oriented processing, or vectorization, as it reduces interpretation overhead. For each vector (batch) of tuples, the DBMS must track the set of

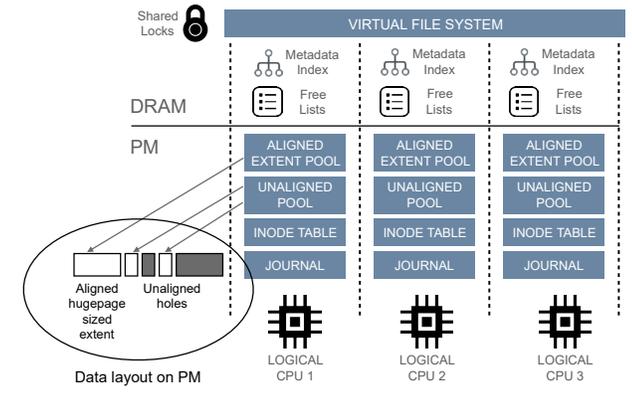
valid (visible) tuples that survive all previous processing steps. To that end, existing systems employ one of two data structures, or filter representations: selection vectors or bitmaps. In this work, we analyze each approach's strengths and weaknesses and offer recommendations on how to implement vectorized operations. Through a wide range of micro-benchmarks, we determine that the optimal strategy is a function of many factors: the cost of iterating through tuples, the cost of the operation itself, and how amenable it is to SIMD vectorization. Our analysis shows that bitmaps perform better for operations that can be vectorized using SIMD instructions and that selection vectors perform better on all other operations due to cheaper iteration logic.

Open Problems In Queueing Theory Inspired By Datacenter Computing.

Mor Harchol-Balter

Queueing Systems, vol. 97, no. 1, 2021, pp. 3-37.

Datacenter operations today provide a plethora of new queueing and scheduling problems. The notion of a “job” has become more general and multi-dimensional. The ways in which jobs and servers can interact have grown in complexity, involving parallelism, speedup functions, precedence constraints, and task graphs. The workloads are vastly more variable and more heavy-tailed. Even the performance metrics of interest are broader than in the past, with multi-dimensional service-level objectives in terms of tail probabilities. The purpose of this article is to expose queueing theorists to new models, while providing suggestions for many specific open problems of interest, as well as some insights into their potential solution.



WineFS Architecture. The figure shows the main components of WineFS. WineFS partitions the file system per logical CPU for concurrency. Each logical CPU has its own journal, inode table, and free lists for aligned extents and holes. WineFS uses DRAM indexes for metadata for efficient directory and resource lookups. The shared locks in the VFS layer help WineFS coordinate its multiple journals.

continued from page 4

Computing” at SC’21, in St. Louis, MO.

- ❖ Mor Harchol-Balter presented “The Most Common Queueing Theory Questions Asked by Computer Systems Practitioners” at TeaPACS ’21, held in conjunction with the IFIP Perf ’21 Conference in Milan, Italy.
- ❖ Benjamin Berg presented “The Case for Phase-Aware Scheduling of Parallelizable Jobs” at IFIP Perf ’21 in Milan, Italy.
- ❖ Jack Kosaian proposed his PhD research on “Reliable and Resource-Efficient Learning Systems via Coding-Theory-Inspired Approaches.”

October 2021

- ❖ Sara McAllister and her colleagues won the Best Paper Award for their work on “Kangaroo: Caching Billions of Tiny Objects on Flash” at SOSP’21.
- ❖ The Mochi Project, contributed to by PDL members George Amvrosiadis, Chuck Cranor and alumni Qing Zheng won a 2021 R&D 100 Award.
- ❖ Huaicheng Li presented “IODA: A Host/Device Co-Design for Strong Predictability Contract on Modern Flash Storage” at SOSP ’21, held virtually in Germany.
- ❖ Ziv Scully presented “The Gittins Policy in the M/G/1 Queue” at WiOpt ’21 in Philadelphia, PA.
- ❖ Rohan Kadekodi, brother of co-author Saurabh Kadekodi presented “WineFS: A Hugepage-aware File System for Persistent Memory that Ages Gracefully” at SOSP ’21.
- ❖ Daehyock Kim defended his PhD research on “Towards Elastic and Resilient In-Network Computing.”

September 2021

- ❖ Graham Gobieski gave his speaking skills talk on “Architectures for Energy-Minimal, On-Device Machine Learning.”
- ❖ Ankur Mallick presented “Rateless Codes for Distributed Non-linear

Computations” at the 2021 Int’l. Symposium on Topics in Coding.

August 2021

- ❖ Lin Ma defended his PhD research on “Self-Driving Database Management Systems: Forecasting, Modeling, and Planning.”
- ❖ Aurick Qiao defended his PhD research on “Elastic Machine Learning Systems with Co-adaptation.”

July 2021

- ❖ Christos Faloutsos was promoted to University Professor at Carnegie Mellon University.
- ❖ Aurick Qiao and his co-authors won Best Paper at OSDI’21 for their work on “Pollux: Co-adaptive Cluster Scheduling for Goodput-Optimized Deep Learning.”
- ❖ Matias Bjørling, collaborating with Abutalib Aghayev, Greg Ganger and George Amvrosiadis presented “ZNS: Avoiding the Block Interface Tax for Flash-based SSDs” at the virtual USENIX 2021 conference.
- ❖ Nathan Beckmann presented “Block-Granularity-Aware Caching” at SPAA ’21.
- ❖ Jack Kosaian presented “Boosting the Throughput and Accelerator Utilization of Specialized CNN Inference Beyond Increasing Batch Size” at ICML ’21.
- ❖ Hongbo Kang, a student of Phil Gibbons, presented “The Processing-in-Memory Model” at SPAA ’21.

June 2021

- ❖ Huaicheng Li has been recognized as a Distinguished Reviewer for his service to Syster 2021
- ❖ Minh S. Q. Truong interned with the Custom Cell Design group at Apple Inc. under Sheela Shreedharan.
- ❖ Xinjing Zhou, a student of Andy Pavlo, presented “Spitfire: A Three-Tier Buffer Manager for Volatile and Non-Volatile Memory” at SIGMOD/PODS ’21.

- ❖ Amadou Ngom presented “Filter Representation in Vectorized Query Execution” at DaMoN ’21, held online in conjunction with SIGMOD/PODS ’21.
- ❖ Tanvir Ahmed Khan, a student of Akshitha Sriraman, presented “Ripple: Profile-Guided Instruction Cache Replacement for Data Center Applications” at ISCA ’21.
- ❖ Daming Dominic Chen defended his PhD research on “Mitigating Memory-Safety Bugs with Efficient Out-of-Process Integrity Checking.”
- ❖ Ziqi Wang proposed his PhD research “On Building a Multiversified Cache Hierarchy with Page Overlays.”
- ❖ Pratik Fegade proposed his PhD research “Compiler Techniques to Optimize Communication for Fragmented Applications.”

May 2021

- ❖ Dana Van Aken, with Andy Pavlo, formed a start-up based on Dana’s PhD research. OtterTune, a play on the once ubiquitous AutoTune, uses machine learning to automatically optimize databases.
- ❖ Charles John McGuffey defended his PhD research on “Modernizing Models and Management of the Memory Hierarchy for Non-Volatile Memory.”
- ❖ Benjamin Berg proposed his PhD research “A Principled Approach to Parallel Job Scheduling.”

April 2021

- ❖ Ravi Teja Mullapudi defended his PhD research on “Dynamic Model Specialization for Efficient Inference, Training and Supervision.”

PDL ALUMNI NEWS

John Linwood Griffin

(PDL 1998-2004)

After eight years at Tripadvisor John switched jobs in July 2021. He is now working remotely from Boston as Head of Acquisition Marketing for Policygenius (www.policygenius.com), a series D startup out of NYC. It's fun work but has strayed pretty far from the cutting edge storage systems research in the PDL!



Vinay Perneti

(PDL 2007-2008)

Vinay recently moved to Meta and is part of the Monitoring & Observability organization. He is supporting teams that are building the time-series, metrics and log platforms used across Meta.

Aditya Sethuraman

(PDL 2007-2008)

Aditya has joined a new startup — Kognitos (www.kognitos.com), and they're building something cool in the HyperAutomation space — a little different from the world of storage and distributed systems!

Daniel Stodolsky

(PDL 1991-1996)

While still at Google, Danner has changed roles. After 7 years with YouTube Video Infrastructure, he has gone back towards his roots, managing the production kernel, borg (scheduling) and performance teams as part of Google Systems and Services Infrastructure teams. Warehouse-scale computing, including joint scheduling of compute, storage and network, are areas of interest, as well as new issues of scale, such as silent data corruption. He and his colleagues also published “Warehouse-scale Video Acceleration: Co-design and Deployment in the Wild” in ASPLOS'21. It is a look at the holistic design considerations behind Google's Warehouse scale video transcoding solution.

Niraj Tolia

(PDL 2002-2007)

Niraj reports he has founded another startup (Alcion) along with a couple of other PDL alumni that were both Greg's students: Eno Thereska (PDL 2002-2007), and Rajat Kateja (PDL 2015-2020). There are other CMU/ECE people involved here too, including Vaibhav Kamra, Niraj's co-founder. The startup is still in stealth mode for now, and we wish them luck in their new venture!

Anand Suresh

(PDL 2011-2012)

Life has been good and hectic for Anand over the past couple of years since the pandemic began. He had his

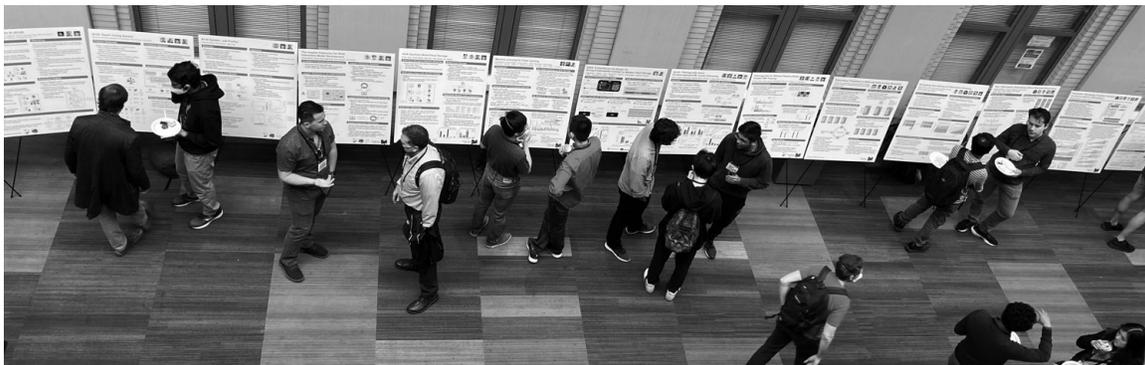


first child, a daughter, has switched jobs twice (he was first at the DFINITY foundation, and is now at Portal Defi working on some Bitcoin/Lightning Network tools). He has moved four times across the US and Canada, finally buying a house and putting down roots in Lafayette CA. He says it has all been a completely crazy sleep-deprived ride!

Ted Wong

(PDL 1997 - 2003)

We are pleased to share that PDL Alum, Ted Wong, is now a celebrity! Ted, whose company of employment, 23andMe, recently went public, had their employees appear on the big screen in Times Square, NY! Ted says he is lucky to be at 23andMe and excited that he gets to work with a lot of talented and fun people.



Visitors to a poster session at the 2022 PDL Spring Visit Day.