AN INFORMAL PUBLICATION FROM ACADEMIA'S PREMIERE STORAGE SYSTEMS RESEARCH CENTER DEVOTED TO ADVANCING THE STATE OF THE ART IN STORAGE AND INFORMATION INFRASTRUCTURES.

CONTENTS

PDL CONSORTIUM MEMBERS

Alibaba Group
Amazon
Datrium
Dell EMC
Facebook
Google
Hewlett Packard Enterprise
Hitachi, Ltd.
IBM Research
Intel Corporation
Micron
Microsoft Research
NetApp, Inc.
Oracle Corporation
Salesforce
Samsung Semiconductor, Inc.
Seagate Technology
Two Sigma
Veritas
Western Digital

PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • SPRING 2019
http://www.pdl.cmu.edu/

Gotta Have HeART: Improving Storage Efficiency by Exploiting Disk-Reliability Heterogeneity

Saurabh Kadekodi, K. V. Rashmi & Gregory Ganger

Large cluster storage systems almost always include a heterogeneous mix of storage devices, even when using devices that are all of the same technology type. Commonly, this heterogeneity arises from incremental deployment and per-acquisition optimization of the makes/models acquired. As a result, a given cluster storage system can easily include several makes/models, each in substantial quantity.

Different makes/models can have substantially different reliabilities, in addition to the well-known differences in capacity and performance. For example, Fig. 1 shows the average annualized failure rates (AFRs) during the useful life (stable operation period) of the 6 HDD makes/models that make up more than 90% of the cluster storage system used for the Backblaze backup service [1]. The highest failure rate is over 3.5X greater than the lowest, and no two are the same. Another recent study has shown that different Flash SSD makes/models similarly exhibit substantial failure rate differences.

Despite such differences, cluster storage redundancy is generally configured as if all of the devices have the same reliability. Unfortunately, this approach leads to configurations that are overly resource-consuming and overly risky. For example, if redundancy settings are configured to achieve a given data reliability target (e.g., a specific mean time to data loss (MTTDL)) based on the highest annualized failure rate (AFR) of any device make/model of any allowed age, then too much space will be used for redundancy associated with data that is stored fully on lower AFR makes/models. If redundancy settings for all data are based on lower AFRs, on the other hand, then data stored fully on higher-AFR devices is not sufficiently protected to achieve the data reliability target.

HeART (Heterogeneity-Aware Redundancy Tuner) is an online tool for guiding exploitation of reliability heterogeneity among disks to reduce the space overhead (and hence the cost) of data reliability. HeART uses failure data observed over time to empirically quantify each disk group's reliability characteristics and determine minimum-capacity redundancy settings that achieve specified target data reliability levels. Studying the Backblaze dataset of 100,000+ HDDs over 5 years, our analysis shows that using HeART's settings could achieve data reliability targets with 11–33% fewer HDDs, depending on the baseline one-scheme-for-all settings. Even when the baseline scheme is a 10-of-14

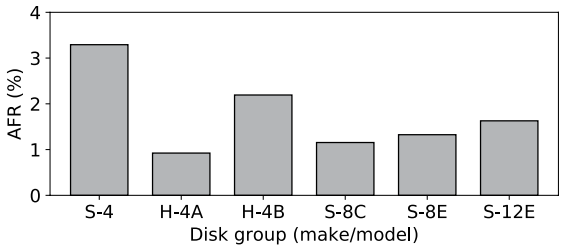


Figure 1: Annualized failure rate (AFR) for the six disk groups that make up >90% of the 100,000+ HDDs used for the Backblaze backup service [1].

continued on page 11

FROM THE DIRECTOR'S CHAIR

GREG GANGER

Hello from fabulous Pittsburgh!

It has been another great year for the Parallel Data Lab. Some highlights include exciting new distributed storage projects, continued growth and success for PDL's storage systems and cloud classes, and lots of great new activities and results in long-standing areas of strength like database systems, ML systems, and cloud infrastructure. Along the way, many students have graduated and joined PDL Consortium companies, PDL researchers have won some big awards, and many cool papers have been published. Specifics can be found throughout the newsletter, but let me highlight a few things.

Since it headlines this newsletter, I'll also lead off my highlights by talking about one of the exciting new research directions arising from the fresh energy brought by the several new faculty who have joined PDL over the past couple of years. (Rashmi, in this case.) The HeART project arose from her insight that there is an opportunity to specialize the erasure codes she loves so much to the differing failure rates of distinct disk groups within large-scale cluster storage. As described in the article, such specialization offers significant potential redundancy overhead reduction, which could reduce dollar costs significantly. We're working together on how to design distributed storage to support adaptive redundancy, which I'm thoroughly enjoying. BTW, her research group is exploring various other ways of exploiting clever erasure-code-based techniques, including how to improve failure-mode performance of Internet content caches and even to reduce tail latency for ML inference systems.

Since I'm bragging on newer PDL faculty, I'll move next to PDL's continuing work on high-performance large-scale storage. (George leads the work in this space.) We continue to explore new metadata scaling approaches, including allowing applications to manage their own namespaces and metadata, for periods of time. In addition to bypassing traditional metadata bottlenecks entirely during the heaviest periods of activity, this approach promises opportunities for efficient in-situ index creation to enable fast queries for subsequent analysis activities. Much of this work is joint with LANL researchers, making it both more fun and more real. There is also a broader effort, with other DoE-sponsored collaborators, to create composable HPC storage systems. And, a recently started effort is exploring how to make cluster storage node software much more efficient, especially on new SMR and zoned-interface technologies.

A lot of exciting database systems stuff is happening at PDL, but perhaps the most exciting is the news that we learned just a few days ago (as I write this): Joy Arulraj won the 2019 ACM SIGMOD Jim Gray Dissertation Award, which is given to the best dissertation in the field of databases. Naturally, his advisor (Andy) has been full of joy… as are we all. Joy's research explored new DBMS architectures for NVM, and was foundational to what is now an important area in both research and practice.

Of course, PDL continues to explore new directions in both database systems and NVM. Much of PDL's expansive database systems research activities center on embedding automation in DBMSs. With an eye toward simplifying administration and improving performance robustness, there are a number of aspects of Andy's overall vision of a self-driving database system being explored and realized. On the NVM front, we have been exploring new approaches to addressing data reliability for direct-access (DAX) NVM and for distributed NVM storage systems, which will be critical to putting VM to use as production storage rather than just cache.

THE PDL PACKET

THE PARALLEL DATA LABORATORY

School of Computer Science
Department of ECE
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891
VOICE 412•268•6716
FAX 412•268•3010

PUBLISHER

Greg Ganger

EDITOR

Joan Digney

The PDL Packet is published once per year to update members of the PDL Consortium. A pdf version resides in the Publications section of the PDL Web pages and may be freely distributed. Contributions are welcome.

THE PDL LOGO

Skibo Castle and the lands that comprise its estate are located in the Kyle of Sutherland in the northeastern part of Scotland. Both 'Skibo' and 'Sutherland' are names whose roots are from Old Norse, the language spoken by the Vikings who began washing ashore regularly in the late ninth century. The word 'Skibo' fascinates etymologists, who are unable to agree on its original meaning. All agree that 'bo' is the Old Norse for 'land' or 'place,' but they argue whether 'ski' means 'ships' or 'peace' or 'fairy hill.'

Although the earliest version of Skibo seems to be lost in the mists of time, it was most likely some kind of fortified building erected by the Norsemen. The present-day castle was built by a bishop of the Roman Catholic Church. Andrew Carnegie, after making his fortune, bought it in 1898 to serve as his summer home. In 1980, his daughter, Margaret, donated Skibo to a trust that later sold the estate. It is presently being run as a luxury hotel.

PARALLEL DATA LABORATORY

FACULTY

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

George Amvrosiadis
David Andersen
Lujo Bauer
Nathan Beckmann
Chuck Cranor
Lorrie Cranor
Christos Faloutsos
Saugata Ghose
Phil Gibbons
Garth Gibson
Mor Harchol-Balter
Gauri Joshi
Todd Mowry
Onur Mutlu
Priya Narasimhan
David O'Hallaron
Andy Pavlo
Majd Sakr
M. Satyanarayanan
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
Alex Glikson
Charlene Zang

VISITING RESEARCHERS / POST DOCS

Rachata Ausavarungnirun
Daniel Berger
Chris Fallin
Hyeontaek Lim
Kazuhiro Saito

GRADUATE STUDENTS

Abutalib Aghayev
V. Parvathi Bhogaraju
Amirali Boroumand
Matt Butrovich
Sol Boucher
Christopher Canel
Chelsea Chen
Dominic Chen
Haoxian Chen
Andrew Chung
Pratik Fegade
Ziqiang Feng
Samarth Gupta
Aaron Harlap
Kevin Hsieh
Fan Hu
Abhilasha Jain
Ankush Jain
Angela Jiang
Ellango Jothimurugesan
Saurabh Kadekodi
Anuj Kalia
Anirudh Kanjani
Rajat Kateja
Thomas Kim
Vamshi Konagari
Jack Kosaian
Michael Kuchnik
Conglong Li
Kunmin Li
Tian Li
Tianyu Li
Yang Li
Elliot Lockerman
Lin Ma
Diptesh Majumdar
Ankur Mallick
Francisco Maturana
Charles McGuffey
Gus Angulo Mezerhane
Prashanth Menon
Yuqing Miao
Wenqi Mou
Yiqun Ouyang
Jun Woo Park
Aurick Qiao
Brian Schwedock
Souptik Sen
Yangjun Sheng
Utsav Sheth
Sivaprasad Sudhir
Aaron Tian
Dana Van Aken
Nandita Vijaykumar
Jianyu Wang
Justin Wang
Ziqi Wang
Jinliang Wei
Daniel Wong
Lin Xiao
Hao Zhang
Huanchen Zhang
Qing Zheng
Giulio Zhou

FROM THE DIRECTOR'S CHAIR

PDL's broad and long-standing focus on the intersection for machine learning (ML) and systems continues to evolve. Several examples of exploring how ML can be applied to make systems more automated and more adaptive are touched on above, and we continue to find cool new places to apply it. We also continue to explore new approaches to system support for large-scale machine learning, recently with a particular focus on new approaches to making DNN training more efficient for large-scale models and for large datasets. We also continue to explore challenges related to training models over geo-distributed data and to exploiting edge resources for high-rate video stream processing applications.

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

Many other ongoing PDL projects are also producing cool results. For example, the Best Student Paper at SoCC'18 was PDL's recent work on Stratus, a scheduler for adaptively-sized "virtual clusters" run in public clouds, exploited job runtime estimates to select, pack, and aggressively release instances to minimize cost. The Best Paper at NSDI'19 was PDL's recent work on efficient datacenter RPCs that showed that excellent RPC performance can be achieved without requiring distributed systems to be highly specialized to niche networking technologies. And, the Best Paper at SIGMOD'18 was PDL's recent work on new data structures for efficient range query filtering. 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.

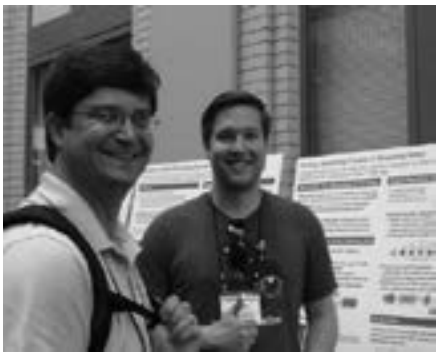
2018 PDL Workshop and Retreat.

May 2019

- 21st annual Spring Visit Day.
- Abutalib Aghayev gave his speaking skills talk on "Evolving Ext4 for Shingled Disks."
- Tianyu Li defended his Masters Thesis "Supporting Hybrid Workloads for In-Memory Database Management Systems via a Universal Columnar Storage Format."
- Anuj Kalia gave his speaking skills talk on "Software-optimized Systems in the Era of Hardware Specialization."
- Jun Woo Park successfully defended his Ph.D. dissertation on "Distribution-based Cluster Scheduling."

April 2019

- Aaron Harlap successfully defended his Ph.D. thesis on "Improving ML Applications in Shared Computing Environments."
- Joy Arulraj won the 2019 SIGMOD Best Dissertation Award for his thesis "The Design and Implementation of a Non-Volatile Memory Database Management System."
- Lorrie Cranor recieved the University of Washington 2019 Alumni Achievement Award.
- Lorrie Cranor named an Andrew Carnegie Fellow.
- Graham Gobieski presented "Intelligence Beyond the Edge: Inference on Intermittent Embedded Systems" at the 24th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'19), in Providence, RI.



Andy Klosterman (PDL Alumni, NetApp, Inc.) discussing research with Chris Canel at the 2018 PDL Spring Visit Day.

March 2019

- Angela Jiang presented her speaking skills talk on "Selective-Backprop: Adaptive Importance Sampling for Training Large Datasets."
- Anuj Kalia and his co-authors received the Best Paper Award for their work on "Datacenter RPCs can be General and Fast" at the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI), in Boston, MA.
- Christopher Canel presented "Scaling Video Analytics on Constrained Edge Nodes" at the 2nd SysML Conference (SysML '19) in Palo Alto, CA.
- Lorrie Cranor was elected to the CRA Board of Directors.

February 2019

- Dong Zhou presented his thesis research on "Data Structure Engineering for High Performance Software Packet Processing."
- Ziqiang Feng gave his speaking skills talk on "Edge-based Discovery of Training Data for Machine Learning."
- Yang Li presented "A Scalable Priority-Aware Approach to Managing Data Center Server Power" at the 25th IEEE International Symposium on High-Performance Computer Architecture, in Washington D.C.
- Rashmi Vinayak gave an invited talk at the 2019 Information Theory and Applications Workshop in San Diego on "Enabling Non-linear Coded Computation via a Learning-based Approach."
- Saurabh Kadekodi presented "Cluster Storage Systems Gotta Have HeART: Improving Storage Efficiency by Exploiting Disk-reliability Heterogeneity" at the 17th USENIX Conference on File and Storage Technologies in Boston, MA.

January 2019

- Lorrie Cranor received the Bosch Distinguished Professorship in Security and Privacy Technologies.
- Lorrie Cranor was named Director of Carnegie Mellon University's CyLab.
- Mor Harchol-Balter received the Joel and Rut Spira Excellence in Teaching Award.
- Nathan Beckmann and Rashmi Vinayak received Google Faculty Research Awards.

December 2018

- Saurabh Kadekodi proposed his Ph.D. research on "Exploiting Heterogeneity in Cluster Storage Systems."
- Huanchen Zhang gave his speaking skills talk on "SuRF: Practical Range Query Filtering with Fast Succinct Tries."
- Mor Harchol-Balter was made an IEEE Fellow.

November 2018

- Conglong Li presented his speaking skills talk on "Workload Analysis and Caching Strategies for Search Advertising Systems."
- Lin Ma presented his speaking skills talk on "The Brain of Databases: Forecasting, Modeling, and Planning for Self-Driving Database Management Systems."
- Gauri Joshi was the recipient of a 2018 IBM Faculty Award.
- Isaac Grosof presented "SRPT for Multiserver Systems Performance Evaluation" at the 36th Inter-

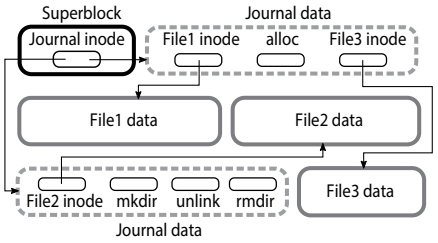
continued on page 31

Reconciling LSM-Trees with Modern Hard Drives using BlueFS

Abutalib Aghayev, Sage Weil, Greg Ganger & George Amvrosiadis

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-19-102, April 2019.

LSM-Trees have become a popular building block in large-scale storage systems where hard drives are the dominant storage medium. Meanwhile, drive makers are shifting to Shingled Magnetic Recording (SMR), a recording technique that increases drive capacity by ~~25%but also works best with a new, backward-incompatible device interface. Large-scale cloud storage providers are updating their proprietary software stacks to utilize SMR drives, but widespread adoption requires more general-purpose support. This paper introduces BlueFS, an open-source user-space file system that allows widely-used LSM-Tree implementations to utilize SMR drives with zero overhead and no code changes. BlueFS's design aggressively specializes data placement and I/O sizes to exposed SMR drive parameters, while hiding those details. As a result, for example, unmodified RocksDB performs random inserts 64% faster atop BlueFS than atop XFS, when storing data on an SMR drive. In addition, LevelDB running on BlueFS is 2–20× faster than GearDB, a recent key-value store designed for SMR drives.



BlueFS data layout. BlueFS is a user-space, extent-based, journaling file system specialized for LevelDB-based LSM-Tree implementations, such as RocksDB, running on a raw HM-SMR drive. BlueFS maintains an inode for each file, with a serial number, modification time, the list of extents allocated to the file, the actual and the allocated size of the file. The superblock resides at a fixed offset in the first conventional zone.

Datacenter RPCs can be General and Fast

Anuj Kalia, Michael Kaminsky & David G. Andersen

16th USENIX Symposium on Networked Systems Design and Implementation (NSDI) Feb. 26–28, 2019, Boston, MA.

It is commonly believed that datacenter networking software must sacrifice generality to attain high performance. The popularity of specialized distributed systems designed specifically for niche technologies such as RDMA, lossless networks, FPGAs, and programmable switches testifies to this belief. In this paper, we show that such specialization is not necessary. eRPC is a new general-purpose remote procedure call (RPC) library that offers performance comparable to specialized systems, while running on commodity CPUs in traditional datacenter networks based on either lossy Ethernet or lossless fabrics. eRPC performs well in three key metrics: message rate for small messages; bandwidth for large messages; and scalability to a large number of nodes and CPU cores. It handles packet loss, congestion, and background request execution. In microbenchmarks, one CPU core can handle up to 10 million small RPCs per second, or send large messages at 75 Gbps. We port a production-grade implementation of Raft state machine replication to eRPC without modifying the core Raft source code. We achieve 5.5 µs of replication latency on lossy Ethernet, which is faster than or comparable to specialized replication systems that use programmable switches, FPGAs, or RDMA.

Fast and Efficient Distributed Matrix-Vector Multiplication Using Rateless Fountain Codes

Ankur Mallick, Malhar Chaudhari & Gauri Joshi

International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 12 - 17 May, 2019 · Brighton, UK.

We propose a rateless fountain coding strategy to alleviate the problem of straggling nodes – computing nodes that unpredictably slowdown or fail – in distributed matrix-vector multiplication. Our algorithm generates linear combinations of the m rows of the matrix, and assigns them to different worker nodes, which then perform row-vector products with the encoded rows. The original matrix-vector product can be decoded as soon as slightly more than m row-vector products are collectively completed by the nodes. This strategy enables fast nodes to steal work from slow nodes, without requiring the knowledge of node speeds. Compared to recently proposed fixed-rate erasure coding strategies which ignore partial work done by straggling nodes, rateless codes have a significantly lower overall delay, and a smaller computational overhead.

Intelligence Beyond the Edge: Inference on Intermittent Embedded Systems

Graham Gobieski, Brandon Lucia & Nathan Beckmann

Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'19), April 13th – April 17th, Providence, RI.

Energy-harvesting technology provides a promising platform for future IoT applications. However, since

continued on page 6

continued from page 5

communication is very expensive in these devices, applications will require inference "beyond the edge" to avoid wasting precious energy on pointless communication. We show that application performance is highly sensitive to inference accuracy. Unfortunately, accurate inference requires large amounts of computation and memory, and energy-harvesting systems are severely resource-constrained. Moreover, energy-harvesting systems operate intermittently, suffering frequent power failures that corrupt results and impede forward progress.

This paper overcomes these challenges to present the first full-scale demonstration of DNN inference on an energy-harvesting system. We design and implement SONIC, an intermittence-aware software system with specialized support for DNN inference. SONIC introduces loop continuation, a new technique that dramatically reduces the cost of guaranteeing correct intermittent execution for loop-heavy code like DNN inference. To build a complete system, we further present GENESIS, a tool that automatically compresses networks to optimally balance inference accuracy and energy, and TAILS, which exploits SIMD hardware available in some microcontrollers to improve energy efficiency. Both SONIC & TAILS guarantee correct intermittent execution without any hand-tuning or performance loss across different power systems. Across three neural networks on a commercially available microcontroller, SONIC & TAILS reduce inference energy by 6.9× and 12.2×, respectively, over the state-of-the-art.

Rateless Codes for Distributed Computations with Sparse Compressed Matrices

Ankur Mallick & Gauri Joshi

IEEE International Symposium on Information Theory (ISIT), July 7-12, 2019, Paris, France.

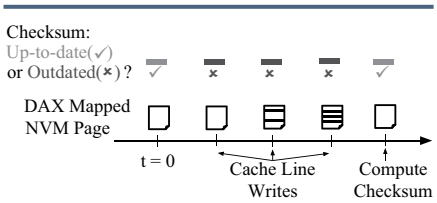
Unpredictable slowdown of worker nodes, or node straggling, is a major bottleneck when performing large matrix computations such as matrix-vector multiplication in a distributed fashion. For sparse matrices, the problem is compounded by irregularities in the distribution of non-zero elements, which leads to an imbalance in the computation load at different nodes. To mitigate the effect of stragglers we use rateless codes that generate redundant linear combinations of the matrix rows (or columns) and distribute them across workers. This coding scheme utilizes all partial work done by worker nodes, and eliminates tail latency. We also propose a balanced row-allocation strategy for allocating rows of a sparse matrix to workers that ensures that equal amount of non-zero matrix entries are assigned to each worker. The entire scheme is designed to work with compressed, memory-efficient formats like CSR/CSC that are used to store sparse matrices in practice. Theoretical analysis and simulations show that our balanced rateless coding strategy achieves significantly lower overall latency than conventional sparse matrix-vector multiplication strategies.

Lazy Redundancy for NVM Storage: Handing the Performance-Reliability Tradeoff to Applications

Rajat Kateja, Andy Pavlo & Greg Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-19-101, April 2019.

Lazy redundancy maintenance can provide direct access non-volatile memory (NVM) with low-overhead data integrity features. The ANON library lazily maintains redundancy (per-page checksums and cross-page parity) for applications that exploit fine-grained direct load/store access to NVM data. To do so, ANON repurposes page table dirty bits to identify pages where redundancy must be updated,



Lazy Checksum Computation Example – By computing per-page checksums lazily, ANON amortizes the computation overhead over multiple cache-line writes to the same NVM page. The reduced computations, and associated performance benefit, come with a window of vulnerability.

addressing the consistency challenges of using dirty bits across crashes. A periodic background thread updates outdated redundancy at a dataset-specific frequency chosen to tune the performance vs. time-to-coverage tradeoff. This approach avoids critical path interpositioning and often amortizes redundancy updates across many stores to a page, enabling ANON to maintain redundancy at just a few percent overhead. For example, MongoDB's YCSB throughput drops by less than 2% when using ANON with a 30 second period and by only 3–7% with a 1 sec period. Compared to the state-of-the-art approach, ANON with a 30 sec period increases the throughput by up to 1.8 for Redis with YCSB workloads and by up to 4.2 for write-only microbenchmarks.

Scaling Video Analytics on Constrained Edge Nodes

Christopher Canel, Thomas Kim, Giulio Zhou, Conglong Li, Hyeontaek Lim, David G. Andersen, Michael Kaminsky & Subramanya R. Dulloor

2nd SysML Conference (SysML '19). March 31-April 2, 2019, Palo Alto, CA.

As video camera deployments continue to grow, the need to process large volumes of real-time data strains wide-area network infrastructure. When

continued on page 7

continued from page 6

per-camera bandwidth is limited, it is infeasible for applications such as traffic monitoring and pedestrian tracking to offload high-quality video streams to a datacenter. This paper presents FilterForward, a new edge-to-cloud system that enables datacenter-based applications to process content from thousands of cameras by installing lightweight edge filters that backhaul only relevant video frames. FilterForward introduces fast and expressive per-application "microclassifiers" that share computation to simultaneously detect dozens of events on computationally-constrained edge nodes. Only matching events are transmitted to the datacenter. Evaluation on two real-world camera feed datasets shows that FilterForward improves computational efficiency and event detection accuracy for challenging video content while substantially reducing network bandwidth use.

Cluster Storage Systems Gotta Have HeART: Improving Storage Efficiency by Exploiting Disk-reliability Heterogeneity

Saurabh Kadekodi, K. V. Rashmi & Gregory R. Ganger

17th USENIX Conference on File and Storage Technologies (FAST '19) Feb. 25–28, 2019 Boston, MA.

Large-scale cluster storage systems typically consist of a heterogeneous mix of storage devices with significantly varying failure rates. Despite such differences among devices, redundancy settings are generally configured in a one-scheme-for-all fashion. In this paper, we make a case for exploiting reliability heterogeneity to tailor redundancy settings to different device groups. We present HeART, an online tuning tool that guides selection of, and transitions between redundancy settings for long-term data reliability, based on observed reliability properties of each disk group. By processing disk failure data over time, HeART identifies the boundaries and steady-state failure rate for each deployed disk group (e.g., by make/model). Using this information, HeART suggests the most space-efficient redundancy option allowed that will achieve the specified target data reliability. Analysis of longitudinal failure data for a large production storage cluster shows the robustness of HeART's failure-rate determination algorithms. The same analysis shows that a storage system guided by HeART could provide target data reliability levels with fewer disks than one-scheme-for-all approaches: 11–16% fewer compared to erasure codes like 10-of-14 or 6-of-9 and 33% fewer compared to 3-way replication.

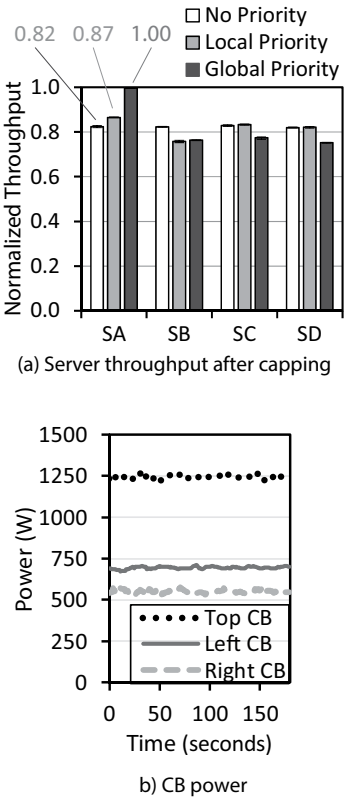
A Scalable Priority-Aware Approach to Managing Data Center Server Power

Yang Li, Charles R. Lefurgy, Karthick Rajamani, Malcolm S. Allen-Ware, Guillermo J. Silva, Daniel D. Heimsoth, Saugata Ghose & Onur Mutlu

HPCA 2019: The 25th International Symposium on High-Performance Computer Architecture, February 16 - 20, 2019, Washington D.C.

Power management is a key component of modern data center design. Power managers must (1) ensure the cost- and energy-efficient utilization of the data center infrastructure, (2) maintain availability of the services provided by the center, and (3) address environmental concerns associated with the center's power consumption. While several power management techniques have been proposed and deployed in production data centers, there are still many challenges to comprehensive data center power management. This is particularly true in public cloud environments, where different jobs have different priority levels, and where high availability is critical.

One example of the challenges facing public cloud data centers involves power capping. As power delivery must be highly reliable and tolerate



(a) Server throughput after power capping policies, normalized to uncapped server throughput; (b) Power at each CB under Global Priority.

wide variation in the load drawn by the data center components, the power infrastructure (e.g., power supplies, circuit breakers, UPS) has high redundancy and overprovisioning. During normal operation (i.e., typical server power demands, and no failures in the center), the power infrastructure is significantly underutilized. Power capping is a common solution to reduce this underutilization, by allowing more servers to be added safely (i.e., without power shortfalls) to the existing power infrastructure, and throttling power consumption in the infrequent cases where the demanded power exceeds the provisioned power capacity to avoid shortfalls. However, state-of-the-art power capping solutions are (1) not directly applicable to

continued on page 20

April 2019
Joy Arulraj Wins SIGMOD Jim Gray Dissertation Award

The Carnegie Mellon Database Group is pleased to announce that CS alumnus Prof. Joy Arulraj (now faculty at the Georgia Institute of Technology) has won the 2019 ACM SIGMOD Jim Gray Dissertation Award. This honor is conferred for the best dissertation in the field of databases for the previous year. Joy's thesis, entitled "The Design and Implementation of a Non-Volatile Memory Database Management Systems", is based on his work exploring new DBMS architectures for NVM. This was work done in collaboration with Intel Labs as part of the Intel Science & Technology Center for Big Data. Joy's research findings were also was published in 2019 as the book "Non-Volatile Memory Database Management Systems" from Morgan & Claypool.

-- Carnegie Mellon University Database Group News, April 28, 2019

March/April 2019
Lorrie Cranor Recipient of Many Honors!

In April, Washington University in St. Louis and the James McKelvey School of Engineering has honored Lorrie Cranor with its 2019 Alumni Achievement Award, praising her passion for teaching and for creating in her wide ranging career an interdisciplinary program leveraging science, humanities and the arts. She is an accomplished scholar and CMU President Farnam Jahanian describes her research as "vigorous, multifaceted and highly relevant to society."

She was also named to the 2019 Class of Andrew Carnegie Fellows by the Carnegie Corporation of New York, a philanthropic foundation that has supported the advancement of education and knowledge for more than a century. She is one of 32 distinguished scholars and writers selected from nearly 300 nominations. "Andrew Carnegie believed in education and understood its influence on the progress of society and mankind. The Andrew Carnegie Fellows Program is an integral part of carrying out the mission he set for our organization," said Vartan Gregorian, president of Carnegie Corporation of New York.

In March, Lorrie was elected to the Computing Research Association board of directors.

-- info from Washington U. News, CMU News, and The Piper, CMU Community News.

March 2019
Best Paper at NSDI'19!

Congratulations to Anuj Kalia, and his co-authors, Michael Kaminsky, and David Andersen for receiving the Best Paper Award for their work on "Datacenter RPCs can be General and Fast" at NSDI'19 (Networked Systems Design and Implementation), which was held in Boston, MA in February. The paper addresses datacenter networking efficiencies for high performance and shows that specialized distributed systems designed specifically for niche technologies such as RDMA, lossless networks, FPGAs, and programmable switches testifies to this belief are not necessary.

January 2019
Mor Harchol-Balter Receives Joel and Rut Spira Excellence in Teaching Award

Congratulations to Mor Harchol-Balter on being recognized for her teaching excellence. The Joel and Ruth Spira Excellence in Teaching Award is presented to an ECE faculty member based on major accomplishments and outstanding qualities and strengths in teaching. Established with an endowment by the Lutron Foundation, this award recognizes individuals who excel in the classroom by helping students learn, understand and apply the fundamentals of engineering.

January 2019
Nathan Beckmann and Rashmi Vinayak Receive Google Faculty Research Awards

Congratulations to Nathan Beckmann and Rashmi Vinayak, both Assistant Professors of Computer Science, on being awarded Google Faculty Research Awards. Rashmi's research focuses on the broad area of computer and networked systems with current attention on reliability, availability, scalability, and performance challenges in data storage and caching systems, in systems for machine learning and in live video streaming. Nathan is interested in computer systems, computer architecture, and performance modeling. His current projects cover hardware, software, and theory and he is currently

continued on page 9

continued from page 8

working on how to build intelligent edge devices, make parallel systems more efficient (particularly by making it less expensive to access data), and use theory to address the huge practical problems still posed by caches.

The Google Faculty Research Awards provide unrestricted gifts as support for research at institutions around the world. The program is focused on funding world-class technical research in Computer Science, Engineering, and related fields.

January 2019
Gauri Joshi Optimizes Computing Systems for IBM's Watson

Machine learning has grown dramatically in engineering and computer science in recent years with the explosion of interest in artificial intelligence. In machine learning, humans — engineers and computer scientists — feed large data sets into a neural network model to train the model to learn from data and eventually identify and analyze patterns and make decisions.

Carnegie Mellon University's Gauri Joshi, and assistant professor of Electrical and Computer Engineering (ECE), is researching the analysis and optimization of computing systems and was named a recipient of a 2018 IBM Faculty Award for her research in distributed machine learning. Faculty Award recipients are nominated by IBM employees in recognition of a specific project that is of significant interest to the company and receive a cash award in support of the selected project.

Joshi's research is about distributing deep learning training algorithms. The datasets used to train neural network models are massive in size, so a single machine is not sufficient to handle the amount of data and the computing required to analyze the data. Therefore, datasets and computations are typically divided across multiple computing nodes, with each node responsible for one part of the data set.

In a distributed machine learning system with data sets divided across nodes, researchers use an algorithm called stochastic gradient descent (SGD), which is at the center of Joshi's research. The algorithm is distributed across the nodes and helps achieve the lowest possible error in the data. It requires exact synchronization, which can lead to delays. Joshi's research strives to strike the best balance between the error and the delay in distributed SGD algorithms. In every iteration of the SGD, a central server is required to communicate with all of the nodes. If any of the nodes slow down, then the entire network slows down to wait for that node, which can significantly reduce the overall speed of the computation. Improving the efficiency and speed of computation are the two main goals of this effort.

Prior to joining Carnegie Mellon's College of Engineering in fall 2017, Joshi was a research staff member at IBM's Thomas J. Watson Research Center.

-- abridged from Carnegie Mellon University News, by Marika Yang, January 9, 2019.

January 2019
Lorrie Cranor Receives the Bosch Chair

CyLab Director Lorrie Cranor has received the Bosch Distinguished Professorship in Security and Privacy Technologies, enabling her to lead a new era of security and privacy research at Carnegie Mellon. The Bosch Chair provides funding support for groundbreaking research addressing important issues related to the security of our connected environment and the privacy of personal data. At the same time, the Bosch Chair affords recognition for the work and career achievements of the CyLab director. "With Lorrie and CyLab as our partner, we at Bosch are looking forward to the opportunity to help shape the future of security and privacy research, together with the world's leading institution on the topic," said Sylvia Vogt, president of the Carnegie Bosch Institute. In addition to serving as the director of CyLab, Cranor is a professor in the Institute for Software Research and in the Department of Engineering and Public Policy, and she serves as co-director of Carnegie Mellon's Privacy Engineering master's degree program.

-- The Piper, CMU Community News

December 2018
Mor Harchol-Balter made an IEEE Fellow

Mor Harchol-Balter has been elevated to fellow status in the Institute of Electrical and Electronics Engineers (IEEE), the world's largest technical professional organization. Fellow status is a distinction reserved for select members who have demonstrated extraordinary accomplishments in an IEEE field of interest. Mor, a professor in CSD since 1999, was cited "for contributions to performance analysis and design of computer systems." Her work on designing new resource-allocation policies includes load-balancing policies, power-management policies and scheduling policies for distributed systems. She is heavily involved in the SIGMETRICS/PERFORMANCE research community and is the author of a popular textbook, "Performance Analysis and Design of Computer Systems."

-- The Piper, CMU Community News, Dec. 12, 2018

continued on page 10

continued from page 9

December 2018
PDL Team Designing Record-breaking Supercomputing File System Framework at Los Alamos National Lab

Trinity occupies a footprint the size of an entire floor of most office buildings, but its silently toiling workers are not flesh and blood. Trinity is a supercomputer at Los Alamos National Laboratory in New Mexico, made up of row upon row of CPUs stacked from the white-tiled floor to the fluorescent ceiling.

The supercomputer is a valuable tool for researchers from a broad range of fields. It can run huge simulations, modeling some of the most complex phenomena known to science. However, continued advances in computing power have raised new issues for researchers.

A team of PDL researchers, including Assistant Professor George Amvrosiadis, Professors Garth Gibson and Greg Ganger, Systems Scientist Chuck Cranor, and Ph.D. student Qing Zheng recently lent a hand to a cosmologist from Los Alamos struggling to simulate complex plasma phenomena. They did not lack the power to run the simulations; rather, Trinity was unable to create and store the massive amounts of data quickly and efficiently. That's where the DeltaFS project came in.

DeltaFS is a file system designed to alleviate the significant burden placed on supercomputers by data-intensive simulations like the cosmologist's plasma simulation. DeltaFS was able to streamline the plasma simulation, bringing what had once been too resource-demanding a task within the supercomputer's capabilities by reorganizing how Trinity processed and moved the data.

First, DeltaFS changed the size and quantity of files the simulation program created. Rather than taking large snapshots encompassing every particle in the simulation (more than a trillion) at once, DeltaFS created a much smaller file for each individual particle. This made it much easier for the scientists to track the activity of individual particles. Through DeltaFS, Trinity was able to create a record-breaking trillion files in just two minutes. Additionally, DeltaFS was able to take advantage of the roughly 10% of simulation time that is usually spent storing the data created, during which Trinity's CPUs are sitting idle. The system tagged data as it flowed to storage and created searchable indices that eliminated hours of time that scientists would have had to spend combing through data manually. This allowed the scientists to retrieve the information they needed 1,000–5,000 times faster than prior methods.

While thrilled with the success of DeltaFS' first real-world test run, the DeltaFS team have already demonstrated a couple of ways that efficiency can be improved, such as indexing or altering file size and quantity. Now they're looking into further ways to take advantage of potential inefficiencies, like using in-process analysis to eliminate unneeded data before it ever reaches storage or compressing information in preparation for transfer to other labs. They even hopes that one day more advanced AI techniques could be incorporated to do much of the observational work performed by scientists, cutting down on observation time and freeing them to focus on analysis and study. But for him and the rest of the DeltaFS team, all of that starts with finding little solutions to improve huge processes.

-- abridged from CMU Engineering News, by Dan Carroll, December 1, 2018.

November 2018
Gauri Joshi Recipient of 2018 IBM Faculty Award

Gauri Joshi, an assistant professor of electrical and computer engineering, has been named a recipient of a 2018 IBM Faculty Award for her research in distributed machine learning. Faculty Award recipients are nominated by IBM employees in recognition of a specific project that is of significant interest to the company and receive a cash award in support of the selected project. Joshi's research is about distributing deep learning training algorithms. "My work is about trying to strike the best balance between the error and the delay in distributed SGD algorithms," Joshi said. "In particular, this framework fits well with the IBM Watson machine learning platform. I will be working with the IBM Watson Machine Learning vision; I will be working with the IBM Research AI team."

-- info from The Piper, CMU Community News, Nov. 1, 2018

continued on page 11

continued from page 10

October 2018
Best Student Paper at SoCC'18!

Congratulations to Andrew Chung and Jun Woo Park, who submitted the Best Student Paper to SoCC '18. Their paper at the Symposium for Cloud Computing, titled "Stratus: Cost-aware Container Scheduling in the Public Cloud," discusses cost considerations of a new cluster scheduler specialized to orchestrate batch job execution on virtual clusters, which dynamically allocates collections of virtual machine instances on public IaaS platforms.

October 2018
Welcome Baby Noah!

The Aghayev family welcomed baby Noah their family on October 9, 2018!

September 2018
PDL Alum Wei Dai Winner of Pittsburgh Business Times 30 under 30 Award!

Wei (David) Dai, who graduated with his Ph.D. in Machine Learning from CMU in 2018 has been listed as one of Pittsburgh's 30 under 30 by the Pittsburgh Business Times. Until recently Wei was the Senior Director of Engineering at Petuum, where they built scalable machine learning platforms for enterprises to easily create and manage complex ML workflows. In February, he took a position as a Senior Machine Learning Engineer Location with Apple in Seattle working on high performance distributed training and on-device / cloud inference frameworks for deep learning.

May 2018
Best Paper at SIGMOD 2018!

The Carnegie Mellon Database Group is pleased to announce that their latest paper "SuRF: Practical Range Query Filtering with Fast Succinct Tries" has won 2018 SIGMOD Best Paper Award. The paper's lead author was CMU CSD Ph.D. Huanchen Zhang. This work was in collaboration with CMU professors Dave Andersen and Andy Pavlo, CMU post-doc Hyeontaek Lim, TUM visiting scholar Viktor Leis, Hewlett Packard Labs' Distinguished Technologist Kimberly Keeton, and Intel Labs' senior research scientist Michael Kaminsky.

GOTTA HAVE HEART

continued from page 1

erasure code whose space-overhead is already low, HeART further reduces disk space used by up to 14%.

HeART uses robust statistical techniques to identify not only a steady-state AFR estimate for each disk group, but also the transitions between deployment stages: infancy —> useful life —> wear-out. HeART assumes that administrators have a baseline redundancy configuration that would be used in HeART's absence; that same configuration should be used for a disk group, when it is initially deployed. HeART then processes failure data for that disk group, during an initial period of months, to determine both when infancy ends and a conservative AFR estimate for the useful life period. It also suggests the most space-efficient redundancy settings supported by the storage system that will achieve the specified data reliability target.

Naturally, the useful life period does not last forever. HeART continues to process failure data for each disk group, automatically identifying the onset of the wear-out period. At this point, a transition to more conservative redundancy (e.g., the original baseline), and possibly decommissioning, is warranted. Importantly, HeART distinguishes between anomalous failure occurrences (e.g., one-time device-independent events, like a power surge, in which many devices fail together) and true changes in the underlying AFR.

The remainder of this article provides more information about the opportunity addressed by and the high-level design of HeART; see our recent FAST paper [2] for more details. Research continues on this exciting new direction, including analysis of larger failure data sets in cooperation with PDL Consortium companies (e.g., NetApp and Google) and integration into HDFS to allow full system experimentation. Stay tuned!

Everyone Should Take HeART

This section makes a case for heterogeneity-aware redundancy tuning. It illustrates potential benefits of using HeART to provide different redundancy schemes for disk groups exhibiting different reliability characteristics in the same commercially used cluster storage system using analyses of an open source dataset from an Internet backup service, Backblaze [1]. This dataset consists of over 5 years of disk reliability statistics from a production cluster storage system with over 100,000 HDDs. We focus on the six make/model disk groups that make

continued on page 12

continued from page 11

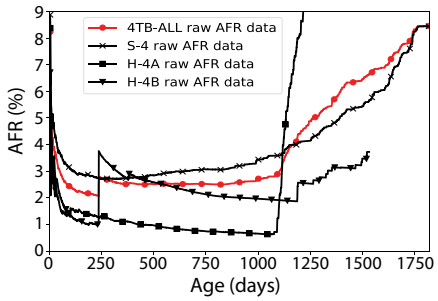


Figure 2: AFR comparison between all 4TB disks grouped together and disk groups broken down by make/model. The AFR differences in make/model-based grouping enables HeART to perform finergrained specialization leading to higher benefits.

up over 90% of the Backblaze deployment, each 8700 disks or more, naming each make/model "X-#y" where "X" indicates make, "#" indicates capacity in TBs, and "y" is an additional letter to uniquify.

We use the standard metric, annualized failure rate (AFR), to describe a disk's fail-stop rate (storage devices can exhibit partial failures and fail-stops or complete failures). As the name suggests, it is the expected percentage of disks that will fail-stop in a given year from a population of disks.

To effectively exploit heterogeneity in AFRs of different disk groups, we want to categorize the disks using parameters that group disks with similar AFRs together and have substantially different AFRs across groups. Each disk group also should have a sizeable population (e.g., ~10,000 or more disks) so that statistical analyses work robustly. Though reasonable groupings may be based on many characteristics (e.g., make/model, capacity, operational conditions, or usage), the Backblaze data only allows us to consider the first two.

Fig. 2 shows the AFR by considering all 4TB disks in the Backblaze dataset as one disk group (red curve with circular marks) and the AFRs of the three make/models of 4TB disks as individual disk groups (black curves). We see significant differences between AFRs when disks are grouped by make/model, suggesting that grouping by capacity is insufficient. HeART groups disks by make/model.

As expected, AFR values of each disk group vary over the lifetime of disks. It is well known that the AFR values over a disk's lifetime follow a bathtub curve. Fig. 3 shows the canonical representation of a disk failure bathtub curve. The lifetime is typically divided into three distinct periods: 1)infant mortality: A higher failure rate in the early days after deployment. This is also called the burn-in period, 2) useful life period: The stable region of operation, where rate of failure is lower, and 3) wearout stage: A higher failure rate towards the disks' end of life due to wear and aging.

Cost Savings. HeART's goal is to reduce storage overhead by tailoring a redundancy scheme accurately representing the failure rate of a disk group during its useful life period. Since infancy and wear-out periods have higher and less stable AFRs compared to useful life, for every disk group, HeART employs the default redundancy scheme (r_{def}) for all infancy and wear-out periods. But, during each disk group's more stable-AFR useful life period, HeART chooses a redundancy scheme that meets the following conditions: (1) it is as reliable as rdef , and (2) it tolerates at least as many failures as rdef. Since most disk groups' AFRs are lower than the worst disk group's AFR, use of lower redundancy levels can often be used without reducing data reliability. Since, large internet services companies try very

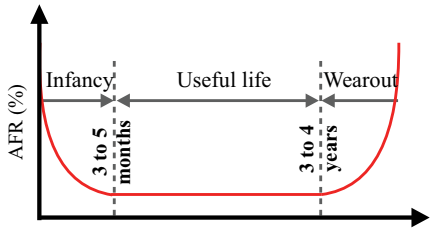


Figure 3: The bathtub curve used to represent disk failure characteristics.

hard to minimize free space (as low as 5%, according to some administrators) in order to minimize capital and operating costs, space savings from lower redundancy levels translate directly into reduced numbers of disks needed... even modest space savings (e.g., 10%) represent a solid case for tailoring redundancy schemes to heterogeneous disk AFRs.

As a concrete example, consider the two oldest makes/models used by Backblaze (S-4 and H-4A), whose useful life AFRs are 3.3% and 0.9% respectively. If one assumes that the S-4 AFR is the worst-case expected for the default redundancy scheme (which is a very conservative assumption, given that wearout failure rates are much worse and tolerated for some time), we observe that redundancy schemes conforming to our two rules above can reduce the storage overhead for H-4A disks by 14-33% depending on the default scheme. Even at the low end, this represents exciting potential.

The Ways of the HeART

Although this short article leaves most design and evaluation details to the full paper [2], this section briefly overviews key challenges to bringing such redundancy tuning into practice and HeART's high-level architecture.

Challenge 1: Function online and be quick. In making our case for HeART, we made use of the complete failure information to identify the 3 stages of a disk group's lifetime and AFR values in each of the stages. In practice, however, AFR values for disk groups deployed in cluster storage systems can only be known in an online fashion. Furthermore, the crux of the cost reduction from HeART comes from quickly tuning the redundancy scheme as soon as we are confident of a disk group having entered its useful life period. Thus, our first challenge in building HeART is that it needs to function in an online fashion taking a continuous stream of disk health

continued on page 13

continued from page 12

data as input and quickly react to the changes in the failure rate.

Challenge 2: Be accurate. It is important to correctly identify the three different stages of the bathtub curve for each disk group. If we are hasty in declaring the end of the infancy period or lax in identifying end of useful life, we might not meet the reliability target because of having tailored the redundancy to a relatively low failure rate during the useful life period. In contrast, if we are too lax about declaring end of infancy or too hasty in declaring onset of the wear-out stage, the opportunity of cost reduction will diminish.

Challenge 3: Filter out anomalies. External events such as power outages, natural disasters or human error can cause large numbers of disks to fail at once. But, such externally-caused bulk failures are not indicative of underlying device failure rates. We must perform AFR anomaly detection to avoid prematurely declaring end of useful life. (As a concrete example, see the AFR spike for H-4B at around 250 days, in Fig. 2.) On the other hand, HeART needs to exercise caution so as to not treat a genuine rise in AFR as an anomaly, which risks not meeting reliability targets.

Architecture. Fig. 4 shows the primary components of HeART. HeART assumes the existence of a disk health monitoring/logging mechanism already in place, which is common in large-scale cluster storage deployments. From the time of deployment till the end of infancy, the default redundancy scheme (rdef) is used to protect the data stored on a disk group. Periodically, disk health data for each disk group is passed through an anomaly detector. Following an anomaly check, the cumulative AFR of every disk group is passed through a change point detector, which checks if a transition to different phase of life has occurred. Once the change point detector announces start of the useful life period, HeART suggests a new redundancy mechanism for the useful life of the disk group (r_{DG}). It computes a determined useful life AFR (AFR_{DG}), which is the AFR at the end of infancy padded with a tunable buffer, and uses it to calculate MTTDLrDG for different redundancy scheme (rDG) options. The buffer is introduced to tolerate the fluctuation of AFR during the useful life period. HeART keeps checking for anomalies and change points throughout the useful life period. When the change point detector marks the end of useful life, HeART raises an alert to reset the redundancy scheme to rdef to handle the increased AFR during wear-out, as was handled in the absence of HeART.

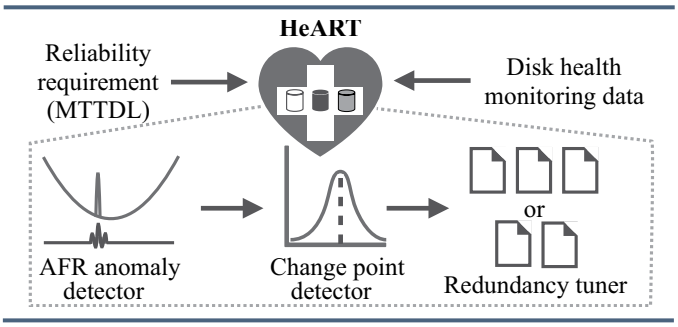


Figure 4: Schematic diagram of HeART. Components include an anomaly detector, an online change point detector, and a redundancy tuner.

HeART suggests redundancy schemes for use with each disk group during its useful life period, enabling safe redundancy tuning based on observed failure data. HeART recommends using the default redundancy scheme employed in the cluster during infancy and wear-out periods. Exploiting HeART's recommendations in a cluster storage system requires some minor data placement policy changes and some online data redistribution.

Data placement. HeART suggests per-disk-group redundancy schemes for hitting a particular data reliability target, based on observed AFRs. To use HeART safely, all data stored using a tailored redundancy scheme must be fully stored within the corresponding disk group—that is, all n chunks (data and parity chunks) of a stripe must be stored on disks within the same group.

Data redistribution. Many cluster storage systems include data redistribution mechanisms to deal with planned decommissioning and capacity/load balancing. Use of HeART will also require their use for transitioning from the default redundancy scheme (r_{def}) to a disk-group-specific scheme (rDG) after infancy, and back again upon onset of wear-out. Although this introduces extra redistribution load, we expect it to have a small impact—at worst, it is two redistributions of the data over the 3–5 year deployment time of the disks.

Bulk changes should not be needed. On its face, HeART's redundancy scheme transitions appear to require massive redistributions all at once, leading to concerns over load spike and capacity: a bulk transition from rDG to the less space-efficient rdef at the end of the useful life period could require more space than is available. Fortunately, we do not expect this issue to arise in practice, as disks of a disk group are deployed over time rather than all at once. Since end of useful life is determined based on deployment age, rolling deployment will mean rolling wear-out.

Overall, we believe that HeART can enable more cost-effective data reliability for cluster storage systems. By robustly estimating per-disk-group AFRs and selecting the best redundancy settings for each, one can avoid the space-inefficiency of one-size-fits-all redundancy schemes offering large potential cost savings. Work on this exciting new research project continues at PDL.

References

[1] Backblaze Disk Reliability Dataset. https://www.backblaze.com/b2/hard-drive-test-data.html.

[2] Cluster Storage Systems Gotta Have HeART: Improving Storage Efficiency by Exploiting Disk-reliability Heterogeneity. S. Kadekodi, K. V. Rashmi, G. R. Ganger. FAST '19, Feb. 25–28, 2019 Boston, MA.

DISSERTATION ABSTRACT: Distribution-based Cluster Scheduling

Jun Woo Park
Carnegie Mellon University, SCS

PhD Defense — May 1, 2019

Modern computing clusters support a mixture of diverse activities, ranging from customer-facing internet services, software development and test, scientific research, and exploratory data analytics. Many schedulers exploit knowledge of pending jobs' runtimes and resource usages as a powerful building block but suffer significant performance penalty if such knowledge is imperfect. This dissertation demonstrates that schedulers that rely on information about job runtimes and resource usages can more robustly address imperfect predictions by looking at likelihoods of possible outcomes rather than single point expected outcomes. This dissertation presents a workload analysis and two case studies of scheduling systems, 3Sigma, and the resource-runtime distribution based scheduler. Characterization of real workloads revealed that there exists inherent variability in the job runtimes and resource usage that cannot be captured by single point estimates. An evaluation of a history-based runtime predictor with four different traces demonstrates it is not trivial to obtain perfect runtime predictions in real workloads, especially if the predictor is provided with insufficient information. 3Sigma is a scheduler that leverages distributions of the relevant runtime histories rather than just a point estimate derived from it. By leveraging distribution and misestimate mitigation mechanisms, 3Sigma is able to make more robust scheduling decisions and outperform state-of-the-art scheduling systems that only rely on limited or no runtime knowledge. The resource-runtime distribution scheduler is a system that can leverage the distribution of resource usage (cpu, memory, and cpu-time) and account the risk of contention to make robust scheduling decisions. The evaluation of the scheduler demonstrates that leveraging full history and mitigation mechanisms allows the scheduler to more robustly address the imperfect predictions and perform almost as good as the hypothetical system equipped with perfect knowledge of runtime and resource usage.

DISSERTATION ABSTRACT: Improving ML Applications in Shared Computing Environments

Aaron Harlap
Carnegie Mellon University, ECE

PhD Defense — April 30, 2019

Machine learning (ML) has become a powerful building block for modern services, scientific endeavors and enterprise processes. The expensive computations required for training ML applications often makes it desirable to run them in a distributed manner in shared computing environments (e.g., Amazon EC2, Microsoft Azure, in-house shared clusters). Shared computing environments introduce a number of challenges, including uncorrelated performance jitter, heterogeneous resources, transient resources and limited bandwidth.

This dissertation demonstrates that, by structuring software frameworks and work distribution to exploit transient resources and address performance jitter and communication bandwidth limitations, we can improve the efficiency of training machine learning models. We support this thesis statement with three case study systems: FlexRR, Proteus, and PipeDream. FlexRR is a distributed machine learning training system that combines a flexible synchronization model with dynamic peer-to-peer re-assignment of work among workers to address stragglers caused by performance jitter. FlexRR observes near ideal run-time, mitigating the adverse effects of stragglers observed in shared computing environments. Proteus is an agile elastic machine learning training system that uses tiers of reliability and intelligent resource management to efficiently utilize transient compute resources. Evaluations on AWS EC2 show that Proteus reduces cost by 85% relative to non-transient pricing, and by 43% relative to previous approaches, while simultaneously reducing runtimes by up to 37%. PipeDream is a distributed training system for deep neural networks (DNNs) that partitions ranges of DNN layers among machines and aggressively pipelines computation and communication. By reducing the amount of communication, and overlapping communication and computation, PipeDream provides a 5x or more improvement in "time to accuracy" for training large DNN models.

DISSERTATION ABSTRACT: The Design and Implementation of a Non-Volatile Memory Database Management System

Joy James Prabhu Arulraj
Carnegie Mellon University, SCS

PhD Defense — July 13, 2018

This dissertation explores the implications of emergent non-volatile memory (NVM) technologies for database management systems (DBMSs). The advent of NVM will fundamentally change the dichotomy between volatile memory and durable storage in DBMSs. These new NVM devices are almost as fast as DRAM, but all writes to it are potentially persistent even after power loss. Existing DBMSs are unable to take full advantage of this technology because their internal architectures are predicated on the assumption that memory is volatile. With NVM, many of the components of legacy DBMSs are unnecessary and will degrade the performance of the data intensive applications.

continued on page 15

continued from page 14

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

DISSERTATION ABSTRACT: Practical Concurrency Testing or: How I Learned to Stop Worrying and Love the Exponential Explosion

Ben Blum
Carnegie Mellon University, SCS

PhD Defense — October 17, 2018

Concurrent programming presents a challenge to students and experts alike because of the complexity of multi-threaded interactions and the difficulty to reproduce and reason about bugs. Stateless model checking is a concurrency testing approach which forces a program to interleave its threads in many different ways, checking for bugs each time. This technique is powerful, in principle capable of finding any nondeterministic bug in finite time, but suffers from exponential explosion as program size increases. Checking an exponential number of thread interleavings is not a practical or predictable approach for programmers to find concurrency bugs before their project deadlines.

In this thesis, I develop several new techniques to make stateless model checking more practical for human use. I have built Landslide, a stateless model checker specializing in undergraduate operating systems class projects. Landslide extends the traditional model checking algorithm with a new framework for automatically managing multiple state spaces according to their estimated completion times, which I show quickly finds bugs should they exist and also quickly verifies correctness otherwise. I evaluate Landslide's suitability for inexpert use by presenting the results of many semesters providing it to students in 15-410, CMU's Operating System Design and Implementation class, and more recently, students in similar classes at the University of Chicago and Penn State University. Finally, I extend Landslide with a new concurrency model for hardware transactional memory, and evaluate several real-world transactional benchmarks to show that stateless model checking can keep up with the developing concurrency demands of real-world programs.

DISSERTATION ABSTRACT: Framework Design for Improving Computational Efficiency and Programming Productivity for Distributed Machine Learning

Jin Kyu Kim
Carnegie Mellon University, SCS

PhD Defense — September 26, 2018

Machine learning (ML) methods are used to analyze data in a wide range of areas, such as finance, e-commerce, medicine, science, and engineering, and the size of machine learning problems has grown very rapidly in terms of data size and model size in the era of big data. This trend drives industry and academic communities toward distributed machine learning that scales out ML training in a distributed system for completion in a reasonable amount of time. There are two challenges in implementing distributed machine

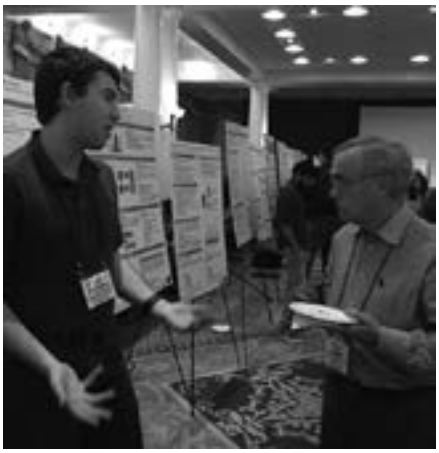
Andy Pavlo, working hard on new database schemes.

learning: computational efficiency and programming productivity. The traditional data-parallel approach often leads to suboptimal training performance in distributed ML due to data dependencies among model parameter updates and nonuniform convergence rates of model parameters. From the perspective of an ML programmer, distributed ML programming requires substantial development overhead even with high-level frameworks because they require an ML programmer to switch to a different mental model for programming from a familiar sequential programming model.

The goal of my thesis is to improve the computational efficiency and programming productivity of distributed machine learning. In an efficiency study, I explore model update scheduling schemes that consider data dependencies and nonuniform convergence speeds of model parameters to maximize convergence per iteration and present a runtime system STRADS that efficiently execute model update scheduled ML applications in a distributed system. In a productivity study, I present familiar sequential-like programming API that simplifies conversion of a sequential ML program into a distributed program without requiring an ML programmer to switch to a different mental for

continued on page 16

continued from page 16



Charles McGuffey describing his research to Paul Suhler (Micron Technology) at a 2018 PDL Retreat poster session.

programming and implement a new runtime system STRADS-Automatic Parallelization(AP) that efficiently executes ML applications written in our API in a distributed system.

THESIS PROPOSAL: Data Structure Engineering for High Performance Software Packet Processing

Dong Zhou, SCS
January 25, 2019

Compared with using specialized hardware, software packet processing on general-purpose hardware provides extensibility and programmability. From software routers to virtual switches to Network Function Virtualization, we are seeing increasing applications of software-based packet processing. However, software-based solutions often face performance challenges, primarily because general-purpose CPUs are not optimized for processing network packets.

We observed that for a wide range of packet processing applications, performance is bottlenecked by one or more data structures. Therefore, this thesis tackles the performance of software packet processing by optimizing the main data structures of the application. To demonstrate the effectiveness of our approach, we examined three applications: Ethernet forwarding, LTE-to-Internet gateway and virtual switches. For each application, we propose algorithmic refinements and engineering principles to improve its main data structures, including:

- A concurrent, read-optimized hash table for x86 platform
- An extremely compact data structure for set separation
- A new cache design that offers both low cache miss rate and high lookup throughput

In all three applications, we are able to achieve higher performance than existing solutions. For example, our Ethernet switch can saturate the maximum number of packets achievable by the underlying hardware, even with one billion FIB entries in the forwarding table.

THESIS PROPOSAL: Exploiting Heterogeneity in Cluster Storage Systems

Saurabh Kadekodi
December 4, 2018

Large-scale storage systems include a heterogeneous mix of storage devices with different reliability, capacity and performance characteristics. This dissertation explores the benefits of explicitly factoring in heterogeneity in the characteristics of storage devices belonging to a single tier, for cheaper redundancy and more load-balanced data placement.

Data reliability in today's cluster storage systems is agnostic to the failure rate differences observed between storage devices in the same tier. Using a production dataset of over 100,000 hard disks, we show that there is a significant diversity in hard disk failure rates, as a function of make/model. We build a case for failure rate tailored redundancy, and design and prototype HeART (Heterogeneity-Aware Redundancy Tuner), an online automated framework to aid the process. HeART continuously monitors the failure rates of different disk groups, and employs robust online algorithms to safely and accurately identify redundancy schemes for each disk group, while meeting a defined reliability target. This leads to lower level of redundancy for devices with low failure rate and more redundancy for devices with high failure rate. Redundancy informed by HeART would result in 13-44% fewer disks than one-scheme-fits-all approaches in the production dataset: 33-44% compared to 3-replication and 13-19% compared to erasure codes like 6-of-9 or 10-of-14.

Technological advancements make capacity heterogeneity also prevalent, and unavoidable in large cluster storage systems. Data placement algorithms agnostic to capacity heterogeneity often cause heavy spindle imbalance across the disk fleet. This load imbalance worsens when there is data heat (accesses per unit time) heterogeneity across different clients. We propose a data placement algorithm that accounts for each dataset's current heat and expected rate of cooling in addition to storage device capacity heterogeneity. These simple heuristics allow for data placement that results in higher storage device utilization, lower rebalancing cost and lower tail latencies.

THESIS PROPOSAL: Efficient and Programmable Distributed Shared Memory Systems for Machine Learning Training

Jinliang Wei, SCS
October 5, 2018

Machine learning training involves frequent and often sparse updates to a large number of numerical values called model parameters. Many distributed training systems have resorted to using distributed shared memory (DSM) (e.g. Parameter Server) for efficient sparse access and in-place updates. Compared to traditional

continued on page 17

continued from page 16

programs, machine learning applications tolerate bounded error, which presents opportunities for trading off learning progress for higher computation throughput. In this thesis, I develop efficient and programmable distributed learning systems, by exploiting this trade-off in the design of distributed shared memory systems, along with parallelization and static and dynamic scheduling.

Thanks to this tolerance to bounded error, a machine learning program can often be parallelized without strictly preserving data dependence. Parallel workers may thus observe inconsistent model parameter values compared to a serial execution. More frequent communication to propagate updates and fresher parameter values may reduce such inconsistency, while incurring higher inter-machine communication overhead. I present a communication management mechanism that automates communication using spare network bandwidth and prioritizes messages according to their importance in order to reduce error due to inconsistency while retaining high computation throughput.

When each model update reads and writes to only a subset of model parameters, it is possible to achieve an efficient parallelization while preserving critical data dependence, exploiting sparse parameter access. Existing systems require substantial programmer effort to take advantage of this opportunity. In order to achieve dependence-preserving parallelization without imposing a huge burden on application programmers, I present a system Orion that provides parallel for-loops on distributed shared memory and parallelizes loops with loop-carried dependence.

At last, I propose to explore dynamic scheduling for dynamic control flow in dataflow systems such as TensorFlow. In TensorFlow, the computation graph is statically partitioned and assigned with computation devices. Static device placement is suboptimal as the operators' load can no longer be determined statically due to dynamic control flow. A suboptimal static device placement may result in imbalanced load and extra communication. It is promising to address the deficiency of static device placement by dynamically scheduling operations based on their load at runtime.

Ellango Jothimurugesan , busy preparing for his talk on "Stochastic Gradient Descent on Streaming Data" at the 2018 PDL Retreat.

THESIS PROPOSAL: Low-Latency, Low-Cost Machine Learning Systems on Large-Scale, Highly-Distributed Data

Kevin Hsieh, ECE
August 9, 2018

The explosive advancement of machine learning (ML) has been the engine of many important applications. The success of an ML-driven application depends on two key factors: low latency and low cost. However, achieving low-latency and low-cost ML is particularly challenging when the ML processes depend on real-world, large-scale data (e.g., user activities, pictures, and videos), which are massive and highly distributed.

In this thesis proposal, we identify three major challenges to achieve low-latency and low-cost ML on massive and highly-distributed data. We describe three research directions that address these challenges with system-level solutions. Our solutions improve the latency and cost of ML on massive and highly-distributed data by one to two orders of magnitude.

First, many ML systems leverage state-of-the-art deep neural networks (DNNs) to process large and continuously growing data (e.g., videos, audios, pictures) with the goal to answer "after the fact" queries such as: identify video frames with objects of certain classes (cars, bags). However, supporting such queries incurs high cost at ingest time or high latency at query time. We present Focus, a system providing both low-cost and low-latency queries over large datasets, using video queries as the case study.

Second, when ML data are highly distributed (e.g., distributed in many data centers across the world), massive communication overhead can drastically slow down an ML system and introduce substantial cost. To this end, we introduce a new, general geo-distributed ML training system, Gaia, that enables efficient communication between data centers by dynamically eliminating insignificant communication while still guaranteeing the correctness of ML algorithms.

THESIS PROPOSAL: Rethinking Cross-layer Abstractions to Enhance Programmability, Portability, and Performance

Nandita Vijaykumar, ECE
August 9th, 2018

The last decades have seen tremendous change and growth across all levels of the computing stack—applications, programming models, compilers, runtime systems, and the hardware architecture. These changes are driven by recent trends, including the push towards domain-specific specialization in hardware and software, consolidation of multiple applications on the same platform via system virtualization, and a new era of data-intensive compu-

continued on page 18

continued from page 17

tation. Programmability, performance portability, and resource efficiency have emerged as critical challenges in harnessing complex and diverse architectures today to obtain high performance and energy efficiency. While there is abundant research, and thus significant improvements, at different levels of the stack that address these very challenges, the interfaces/abstractions between the levels of the computing stack have largely remained the same.

This thesis makes a case for rethinking the cross-layer abstractions in the new landscape of fast-evolving hardware and software. While today the cross-layer abstractions are primarily designed for program functionality and correctness, we explore how richer interfaces can make a significant difference in how we optimize for programmability, performance portability, and resource efficiency across the computing stack. We propose 4 different approaches to designing richer abstractions between the application, system software, and hardware architecture: (i) Expressive Memory: A unifying cross-layer abstraction to express and communicate higher-level program semantics from the application to the underlying system/architecture to enhance memory optimization; (ii) The Locality Descriptor: A cross-layer abstraction to express and exploit data locality in GPUs; (iii) Zorua: A framework to decouple the programming model from management of on-chip resources and parallelism in GPUs; (iv) Assist Warps: A helper-thread abstraction to dynamically leverage underutilized compute/memory bandwidth in GPUs to perform useful work. We describe each concept and propose the research questions to be addressed in this thesis.

PDL Alumni reunion at FAST '19 in Boston in February. From l to r, Ippokratis Pandis, John Strunk, John Griffin, Greg Ganger and Rajat Kateja.

THESIS PROPOSAL: Distribution-based cluster scheduling

Jun Woo Park, SCS
May 16, 2018

This thesis seeks to propose and evaluate a scheduler that can leverage full distributions (e.g.,the histogram of observed runtimes or resource usage) rather than single point estimates. Knowing point estimates, such as how long each job will execute, enables a scheduler to more effectively pack jobs with diverse time concerns (e.g., deadline vs. the-sooner-the-better) and placement preferences on heterogeneous cluster resources. But, existing schedulers use single-point estimates (e.g., mean or median of a relevant subset of historical runtimes), and we show that they are fragile in the face of real-world estimate error profiles. In particular, analysis of job traces from three different large-scale cluster environments shows that, while the runtimes of many jobs can be predicted well, even state-of-the-art predictors have wide error profiles with 8-23% of predictions off by a factor of two or more. Instead of reducing relevant history to a single point, a distribution provides much more information (e.g., variance, possible multi-modal behaviors, etc.) and allows the scheduler to make more robust decisions. By considering the range of possible runtimes and resource usage for a job, and their likelihoods, the scheduler can explicitly consider various potential outcomes from each possible scheduling option and select an option based on optimizing the expected outcome.

THESIS PROPOSAL: Improving ML Applications in Shared Computing Environments

Aaron Harlap, ECE
May 16, 2018

Statistical machine learning (ML) has become a powerful building block for modern services, scientific endeavors and enterprise processes. We focus on the major subset of ML approaches that employ iterative algorithms to determine model parameters that best fit a given set of input data. Such algorithms iterate over the input data, refining their current best estimate of the parameter values to converge on a final solution.

The expensive computations required for training ML applications often makes it desirable to run them in a distributed manner in shared computing environments (e.g. Amazon EC2, Microsoft Azure, in-house shared clusters). Distributed training of ML applications commonly require the resources involved to maintain parameter data (solution state), evenly distribute work, synchronize progress and communicate amongst each other in order for the ML application to function effectively.

Shared computing environments introduce a number of challenges including uncorrelated performance jitter, heterogeneous resources, transient resources and limited bandwidth. In our work we focus on improving the efficiency, reducing cost and reducing runtime of training ML applications in shared computing environments by addressing the challenges described.

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

Huanchen Zhang,CS
April 30, 2018

This thesis seeks to address the challenge of building space-efficient yet high- per-

continued on page 19

continued from page 18

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

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

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

THESIS PROPOSAL: Efficient Networked Systems for Datacenter Fabrics with RPCs

Anuj Kalia, SCS
March 23, 2018

Datacenter networks have changed radically in recent years. Their bandwidth and latency has improved by orders of magnitude, and advanced network devices such as NICs with Remote Direct Memory Access (RDMA) capabilities and programmable switches have been deployed. The conventional wisdom is that to best use fast datacenter networks, distributed systems must be redesigned to offload processing from server CPUs to network devices. In this dissertation, we show that conventional, non-offloaded designs offer better or comparable performance for a wide range of datacenter workloads, including key-value stores, distributed transactions, and highly-available replicated services.

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

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

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

Larry Rudolph of Two Sigma, talks about Security In the Cloud during the PDL Consortium Speaker Series at visit day in 2018.

MASTERS THESIS: Supporting Hybrid Workloads for In-Memory Database Management Systems via a Universal Columnar Storage Format

Tianyu Li, SCS
May 6th, 2019

The proliferation of modern data processing ecosystems has given rise to open-source columnar data file formats. The key advantage of these formats is that they allow organizations to load data from database management systems (DBMSs) once instead of having to convert it to a new format for each usage. These formats, however, are read-only. This means that organizations must still use a heavy-weight transformation process to load data from their original format into the desired columnar format. We aim to reduce or even eliminate this process by developing an in-memory storage management architecture for transactional DBMSs that is aware of the eventual usage of its data and operates directly on columnar storage blocks. We introduce relaxations to common analytical formats requirements to efficiently update data in blocks, and rely on a lightweight in-memory transformation process to convert blocks back to analytical forms when they are cold . We also describe how to directly access data from third-party analytical tools via remote procedure calls with minimal serialization overhead. To evaluate our work, we implemented our storage engine based on the Apache Arrow format and integrated it into CMU's new DBMS. Our experiments show that our approach achieves comparable performance with dedicated OLTP DBMS while also enabling significantly faster data exports to external data science and machine learning libraries than existing approaches.

continued from page 7

the redundant power infrastructure used in highly-available data centers; and (2) oblivious to differing workload priorities across the entire center when power consumption needs to be throttled, which can unnecessarily slow down high-priority work.

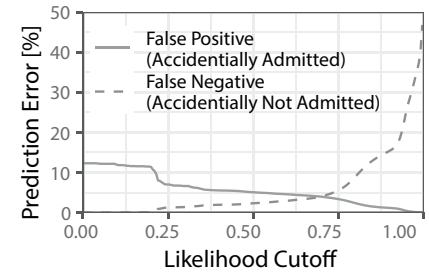
To address this need, we develop CapMaestro, a new power management architecture with three key features for public cloud data centers. First, CapMaestro is designed to work with multiple power feeds (i.e., sources), and exploits server-level power capping to independently cap the load on each feed of a server. Second, CapMaestro uses a scalable, global priority-aware power capping approach, which accounts for power capacity at each level of the power distribution hierarchy. It exploits the underutilization of commonly-employed redundant power infrastructure at each level of the hierarchy to safely accommodate a much greater number of servers. Third, CapMaestro exploits stranded power (i.e., power budgets that are not utilized) in redundant power infrastructure to boost the performance of workloads in the data center. We add CapMaestro to a real cloud data center control plane, and demonstrate the effectiveness of all three key features. Using a large-scale data center simulation, we demonstrate that CapMaestro significantly and safely increases the number of servers for existing infrastructure. We also call out other key technical challenges the industry faces in data center power management.

Towards Lightweight and Robust Machine Learning for CDN Caching

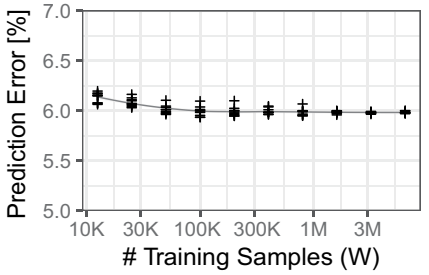
Daniel S. Berger

HotNets-XVII, November 15–16, 2018, Redmond, WA, USA.

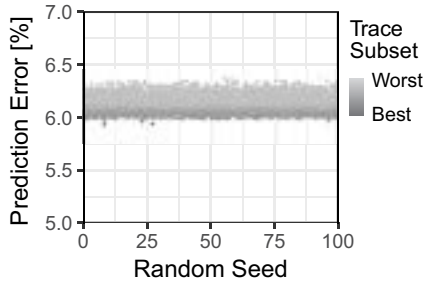
Recent advances in the field of reinforcement learning promise a general approach to optimize networking systems. This paper argues against the recent trend for generalization by introducing a case study where domain-specific modeling enables the application of lightweight and robust learning techniques. We study CDN caching systems, which make a good case for optimization as their performance directly affects operational costs, while currently relying on many hand-tuned parameters. In caching, reinforcement learning has been shown to perform suboptimally when compared to simple heuristics. A key challenge is that rewards (cache hits) manifest with large delays, which prevents timely feedback to the learning algorithm and introduces significant complexity. This paper shows how to significantly simplify this problem by explicitly modeling optimal caching decisions (OPT). While prior work considered deriving OPT impractical, recent theoretical modeling advances change this assumption. Modeling OPT enables even lightweight decision trees to outperform state-of-the-art CDN caching heuristics.



(a) False Positives/Negatives



(b) Impact of Training Set Size



(c) Impact of Random Seeds

Preliminary results on the accuracy of our proposal, LFO, measured in terms of prediction error (requests where OPT and LFO's prediction disagree). (a) The false positive rate and false negative rate depend on the cutoff parameter, but are roughly stable between .25 and .75. (b) The prediction error quickly decays with increasing training set size and stabilizes after around 60K samples. (c) Random seeds, trace subsets, and hyperparameters have only a small impact on LFO's accuracy.

SRPT for Multiserver Systems

Isaac Grosof, Ziv Scully & Mor Harchol-Balter

Performance Evaluation , vol. 127-128, Nov. 2018, pp. 154-175. Also appeared in the following conference: 36th International Symposium on Computer Performance, Modeling, Measurements, and Evaluation (Performance 2018), Toulouse, France, December 2018. Best Student Paper Award!

The Shortest Remaining Processing Time (SRPT) scheduling policy and its variants have been extensively studied in both theoretical and practical settings. While beautiful results are known for single-server SRPT, much less is known for multiserver SRPT. In particular, stochastic analysis of the M/G/k under SRPT is entirely open. Intuition suggests that multiserver SRPT should be optimal or near-optimal for minimizing mean response time. However, the only known analysis of multiserver SRPT is in the worst-case adversarial setting, where SRPT can be far from optimal. In this paper, we give the first stochastic analysis bounding mean response time of the M/G/k under SRPT. Using our response time bound, we show that multiserver SRPT has asymptoti-

continued on page 21

continued from page 20

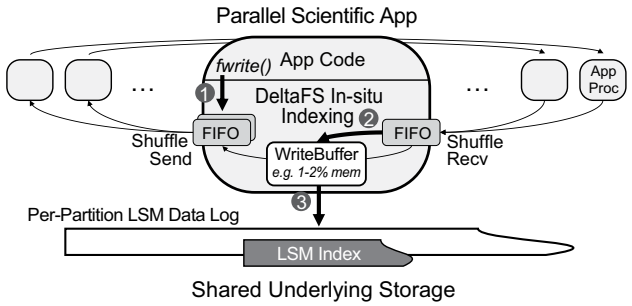
cally optimal mean response time in the heavy-traffic limit. The key to our bounds is a strategic combination of stochastic and worst-case techniques. Beyond SRPT, we prove similar response time bounds and optimality results for several other multiserver scheduling policies.

Scaling Embedded In-Situ Indexing with DeltaFS

Qing Zheng, Charles D. Cranor, Danhao Guo, Gregory R. Ganger, George Amvrosiadis, Garth A. Gibson, Bradley W. Settlemyer, Gary Grider & Fan Guo

SC18, November 11-16, 2018, Dallas, Texas, USA.

Analysis of large-scale simulation output is a core element of scientific inquiry, but analysis queries may experience significant I/O overhead when the data is not structured for efficient retrieval. While in-situ processing allows for improved time-to-insight for many applications, scaling in-situ frameworks to hundreds of thousands of cores can be difficult in practice. The DeltaFS in-situ indexing is a new approach for in-situ processing of massive amounts of data to achieve efficient point and small-range queries. This paper describes the challenges and lessons learned when scaling this in-situ processing function to hundreds of thousands of cores. We propose techniques for scalable all-to-all communication that is memory and bandwidth efficient, concurrent indexing, and specialized LSM-Tree formats. Combining these techniques allows DeltaFS to control the cost of in-situ processing while maintaining 3 orders of magnitude query speedup when scaling alongside the popular VPIC particle-in-cell code to 131,072 cores.



DeltaFS in-situ indexing is library code linked into the processes of a parallel job. Data written by the job is first partitioned and shuffled to the process responsible for it (Step 1). Then, the data is received at the other end (Step 2), and indexed using a modified LSM-Tree (Step 3).

Focus: Querying Large Video Datasets with Low Latency and Low Cost

Kevin Hsieh, Ganesh Ananthanarayanan, Peter Bodik, Shivaram Venkataraman, Paramvir Bahl, Matthai Philipose, Phillip B. Gibbons & Onur Mutlu

13th USENIX Symposium on Operating Systems Design and Implementation (OSDI) Oct. 8–10, 2018, Carlsbad, CA.

Large volumes of video are continuously recorded by cameras deployed for traffic control and surveillance with the goal of answering "after the fact" queries such as: identify video frames with objects of certain classes (cars, bags) from many days of recorded video. Current systems for processing such queries on large video datasets incur either high cost at video ingest time or high latency at query time. We present Focus, a system providing both low-cost and low-latency querying on large video datasets. Focus' architecture flexibly and effectively divides the query processing work between ingest time and query time. At ingest time (on live videos), Focus uses cheap convolutional network classifiers (CNNs) to construct an approximate index of all possible object classes in each frame (to handle queries for any class in the future). At query time, Focus leverages this approximate index to provide low latency, but compensates for the lower accuracy of the cheap CNNs through the judicious use of an expensive CNN. Experiments on commercial video streams show that Focus is 48 (up to 92) cheaper than using expensive CNNs for ingestion, and provides 125 (up to 607) lower query latency than a state-of-the-art video querying system (NoScope).

SOAP Bubbles: Robust Scheduling Under Adversarial Noise

Ziv Scully & Mor Harchol-Balter

56th Annual Allerton Conference on Communication, Control, and Computing, 2-5 Oct. 2018. Monticello, IL.

A great many scheduling policies for the M/G/1 queue are so-called SOAP policies [1], meaning they assign each job a priority based on its age, the amount of service it has received so far. Perhaps the most notable example is the Gittins policy, which minimizes mean response time when job sizes are unknown. However, in some computer systems even job ages, let alone job sizes, are not precisely known by the scheduler. This can occur when scheduling in a time-shared system or over a network. Given that the Gittins policy relies on knowing exact job ages, it is not clear how to minimize mean response time in such settings. In this paper we study scheduling for the M/G/1 when the scheduler knows only approximate job ages. We find that naively using the traditional Gittins policy is not robust, meaning that introducing even an infinitesimal amount of noise in job ages can cause a large jump in mean response time. By examining the ways in which this naive policy fails, we construct a simple variation of the Gittins policy, called the shift-flat Gittins policy, which is indeed robust to noise and therefore has near-optimal mean response time.

continued on page 22

continued from page 21

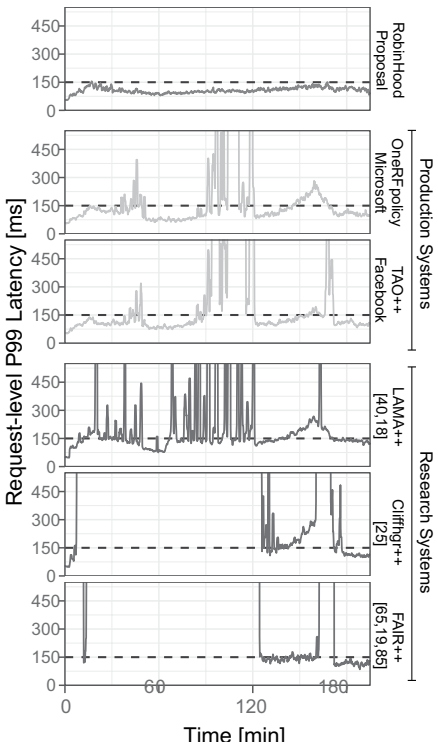
Moreover, we show that our shift-flat construction generalizes, yielding a robust variation of any SOAP policy.

RobinHood: Tail Latency Aware Caching—Dynamic Reallocation from Cache-Rich to Cache-Poor

Daniel S. Berger, Benjamin Berg, Timothy Zhu, Siddhartha Sen & Mor Harchol-Balter

13th USENIX Symposium on Operating Systems Design and Implementation (OSDI '18). October 8–10, 2018, Carlsbad, CA, USA.

Tail latency is of great importance in user-facing web services. However, maintaining low tail latency is challenging, because a single request to a web application server results in multiple queries to complex, diverse backend services (databases, recommender systems, ad systems, etc.). A request is not complete until all of its queries have completed. We analyze a Microsoft production system and find that backend query latencies vary by more than two orders of magnitude across backends and over time, resulting in high request tail latencies.



Comparison of the P99 request latency of RobinHood, two production caching systems, and three state-of-the-art research caching systems, which we emulated in our testbed. All systems are subjected to three load spikes. We draw a dashed line at 150ms, which is the worst latency under RobinHood.

We propose a novel solution for maintaining low request tail latency: repurpose existing caches to mitigate the effects of backend latency variability, rather than just caching popular data. Our solution, RobinHood, dynamically reallocates cache resources from the cache-rich (backends which don't affect request tail latency) to the cache-poor (backends which affect request tail latency). We evaluate RobinHood with production traces on a 50- server cluster with 20 different backend systems. Surprisingly, we find that RobinHood can directly address tail latency even if working sets are much larger than the cache size. In the presence of load spikes, RobinHood meets a 150ms P99 goal 99.7% of the time, whereas the next best policy meets this goal only 70% of the time.

Exploiting Locality in Graph Analytics through Hardware-Accelerated Traversal Scheduling

Anurag Mukkara, Nathan Beckmann, Maleen Abeydeera, Xiaosong Ma & Daniel Sanchez

51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 20-24 Oct. 2018, Fukuoka, Japan.

Graph processing is increasingly bottlenecked by main memory accesses. On-chip caches are of little help because the irregular structure of graphs causes seemingly random memory references. However, most real-world graphs offer significant potential locality—it is just hard to predict ahead of time. In practice, graphs have well-connected regions where relatively few vertices share edges with many common neighbors. If these vertices were processed together, graph processing would enjoy significant data reuse. Hence, a graph's traversal schedule largely determines its locality.

This paper explores online traversal scheduling strategies that exploit the community structure of real-world graphs to improve locality. Software graph processing frameworks use simple, locality-oblivious scheduling because, on general-purpose cores, the benefits of locality-aware scheduling are outweighed by its overheads. Software frameworks rely on offline preprocessing to improve locality. Unfortunately, preprocessing is so expensive that its costs often negate any benefits from improved locality. Recent graph processing accelerators have inherited this design. Our insight is that this misses an opportunity: Hardware acceleration allows for more sophisticated, online locality-aware scheduling than can be realized in software, letting systems significantly improve locality without any preprocessing.

To exploit this insight, we present bounded depth-first scheduling (BDFS), a simple online locality-aware scheduling strategy. BDFS restricts each core to explore one small, connected region of the graph at a time, improving locality on graphs with good community structure. We then present HATS, a hardware-accelerated traversal scheduler that adds just 0.4% area and 0.2% power over general-purpose cores.

We evaluate BDFS and HATS on several algorithms using large real-world graphs. On a simulated 16-core system, BDFS reduces main memory accesses by up to 2.4× and by 30% on average. However, BDFS is too expensive in

continued on page 23

continued from page 22

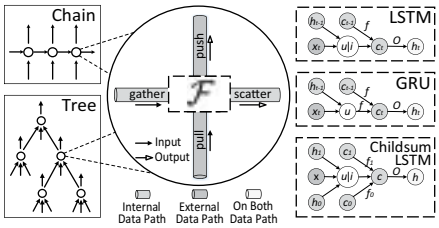
software and degrades performance by 21% on average. HATS eliminates these overheads, allowing BDFS to improve performance by 83% on average (up to 3.1×) over a locality-oblivious software implementation and by 31% on average (up to 2.1×) over specialized prefetchers.

Cavs: An Efficient Runtime System for Dynamic Neural Networks

Shizhen Xu, Hao Zhang, Graham Neubig, Wei Dai, Jin Kyu Kim, Zhijie Deng, Qirong Ho, Guangwen Yang & Eric P. Xing

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

Recent deep learning (DL) models are moving more and more to dynamic neural network (NN) architectures, where the NN structure changes for every data sample. However, existing DL programming models are inefficient in handling dynamic network architectures because of: (1) substantial overhead caused by repeating dataflow graph construction and processing every example; (2) difficulties in batched execution of multiple samples; (3) inability to incorporate graph optimization techniques such as those used in static graphs. In this paper, we present "Cavs", a runtime system that overcomes these bottlenecks and achieves efficient training and inference of dynamic NNs. Cavs represents a dynamic NN as a static vertex function F and a dynamic instance-specific graph G. It avoids the overhead of repeated graph construction by only declaring and constructing F once, and allows for the use of static graph optimization techniques on pre-defined operations in F. Cavs performs training and inference by scheduling the execution of F following the dependencies in G, hence naturally exposing batched execution opportunities over different samples. Experiments comparing Cavs to state-of-the-art frameworks for dynamic



Cavs represents a dynamic structure as a dynamic input graph G (left) and a static vertex function F (right).

NNs (TensorFlow Fold, PyTorch and DyNet) demonstrate the efficacy of our approach: Cavs achieves a near one order of magnitude speedup on training of dynamic NN architectures, and ablations verify the effectiveness of our proposed design and optimizations.

The Parallel Persistent Memory Model

Guy E. Blelloch, Phillip B. Gibbons, Yan Gu, Charles McGuffey & Julian Shun

SPAA '18, July 16–18, 2018, Vienna, Austria.

We consider a parallel computational model, the Parallel Persistent Memory model, comprised of P processors, each with a fast local ephemeral memory of limited size, and sharing a large persistent memory. The model allows for each processor to fault at any time (with bounded probability), and possibly restart. When a processor faults, all of its state and local ephemeral memory is lost, but the persistent memory remains. This model is motivated by upcoming non-volatile memories that are nearly as fast as existing random access memory, are accessible at the granularity of cache lines, and have the capability of surviving power outages. It is further motivated by the observation that in large parallel systems, failure of processors and their caches is not unusual.

We present several results for the model, using an approach that breaks a computation into capsules, each of which can be safely run multiple times. For the single-processor version we describe how to simulate any program in the RAM, the external memory model, or the ideal cache model with an expected constant factor overhead. For the multiprocessor version we describe how to efficiently implement a work-stealing scheduler within the model such that it handles both soft faults, with a processor restarting, and hard faults, with a processor permanently failing. For any multithreaded fork-join computation that is race free, write-after-read conflict free and has W work, D depth, and C maximum capsule work in the absence of faults, the scheduler guarantees a time bound on the model of $O(W/P_A + DP/P_A \lceil \log_{1/(Cf)} W \rceil)$ in expectation, where P is the maximum number of processors, P_A is the average number, and $f \leq 1/(2C)$ is the probability a processor faults between successive persistent memory accesses. Within the model, and using the proposed methods, we develop efficient algorithms for parallel prefix sums, merging, sorting, and matrix multiply.

Stratus: Cost-aware Container Scheduling in the Public Cloud

Andrew Chung, Jun Woo Park & Gregory R. Ganger

ACM Symposium on Cloud Computing, 2018 (SoCC'18), Carlsbad, CA October 11-13, 2018.

Stratus is a new cluster scheduler specialized for orchestrating batch job execution on virtual clusters, dynamically allocated collections of virtual machine instances on public IaaS platforms. Unlike schedulers for conventional clusters, Stratus focuses primarily on dollar cost considerations, since public clouds provide effectively unlimited, highly heterogeneous resources allocated on demand. But, since resources are charged-for while allocated, Stratus aggressively

continued on page 24

continued from page 23

packs tasks onto machines, guided by job runtime estimates, trying to make allocated resources be either mostly full (highly utilized) or empty (so they can be released to save money). Simulation experiments based on cluster workload traces from Google and Two Sigma show that Stratus reduces cost by 17–44% compared to state-of-the-art approaches to virtual cluster scheduling.

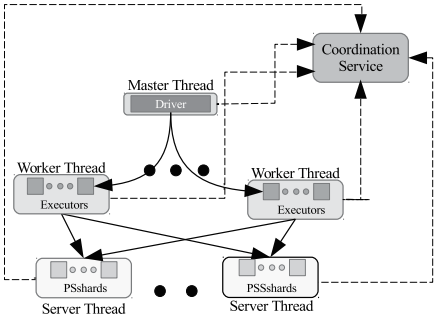
Litz: Elastic Framework for High-Performance Distributed Machine Learning

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

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

Machine Learning (ML) is an increasingly popular application in the cloud and data-center, inspiring new algorithmic and systems techniques that leverage unique properties of ML applications to improve their distributed performance by orders of magnitude. However, applications built using these techniques tend to be static, unable to elastically adapt to the changing resource availability that is characteristic of multi-tenant environments. Existing distributed frameworks are either inelastic, or offer programming models which are incompatible with the techniques employed by high-performance ML applications.

Motivated by these trends, we present Litz, an elastic framework supporting distributed ML applications. We categorize the wide variety of techniques employed by these applications into three general themes — stateful workers, model scheduling, and relaxed consistency — which are collectively supported by Litz's programming model. Our implementation of Litz's execution system transparently enables elasticity and low-overhead execution. We implement several popular ML applications using Litz, and show that



High-level architecture of Litz. The driver in the master thread dispatches micro-tasks to be performed by executors on the worker threads. Executors can read and update the global model parameters distributed across PSshards on the server threads.

they can scale in and out quickly to adapt to changing resource availability, as well as how a scheduler can leverage elasticity for faster job completion and more efficient resource allocation. Lastly, we show that Litz enables elasticity without compromising performance, achieving competitive performance with state-of-the-art non-elastic ML frameworks.

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

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

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

File system performance on modern primary storage devices (Flash-based SSDs) is greatly affected by aging of the free space, much more so than were mechanical disk drives. We introduce Geriatrix, a simple-to-use profile driven file system aging tool that induces target levels of fragmentation in both allocated files (what you see) and remaining free space (what you don't see), unlike previous approaches that focus on just the former. This paper describes and evaluates the effectiveness of Geriatrix, showing that it recreates both fragmentation effects better than previous approaches. Using Geriatrix, we show that measurements presented in many recent file systems papers are higher than should be expected, by up to 30% on mechanical (HDD) and up to 80% on Flash (SSD) disks. Worse, in some cases, the performance rank ordering of file system designs being compared are different from the published results. Geriatrix will be released as open source software with eight built-in aging profiles, in the hopes that it can address the need created by the increased performance impact of file system aging in modern SSD-based storage.

Putting the "Micro" Back in Microservice

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

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

Modern cloud computing environments strive to provide users with fine-grained scheduling and accounting, as well as seamless scalability. The most recent face to this trend is the "serverless" model, in which individual functions, or microservices, are executed on demand. Popular implementations of this model, however, operate at a relatively coarse granularity, occupying resources for minutes at a time and requiring hundreds of milliseconds for a cold launch. In this paper, we describe a novel design for providing "functions as a service" (FaaS) that attempts to be truly micro: cold launch times in microseconds that enable even finer-grained resource accounting and support latency-critical applications. Our proposal is to eschew much of the traditional serverless infrastructure in favor of language-based isolation. The result is microsecond-granularity launch latency, and microsecond-scale

continued on page 25

continued from page 24

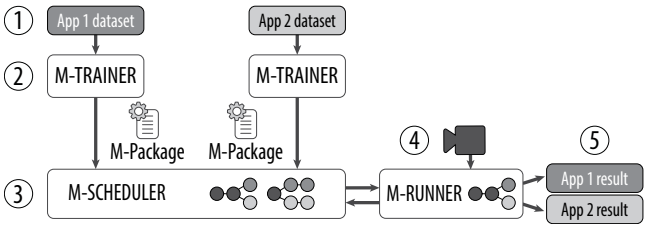
preemptive scheduling using high-precision timers.

Mainstream: Dynamic Stem-Sharing for Multi-Tenant Video Processing

Angela H. Jiang, Daniel L.K. Wong, Christopher Canel, Lilia Tang, Ishan Misra, Michael Kaminsky, Michael A. Kozuch, Padmanabhan Pillai, David G. Andersen & Gregory R. Ganger

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

Mainstream is a new video analysis system that jointly adapts concurrent applications sharing fixed edge resources to maximize aggregate result quality. Mainstream exploits partial-DNN (deep neural network) compute sharing among applications trained through transfer learning from a common base DNN model, decreasing aggregate per-frame compute time. Based on the available resources and mix of applications running on an edge node, Mainstream automatically determines at deployment time the right trade-off between using more specialized DNNs to improve per-frame accuracy, and keeping more of the unspecialized base model to increase sharing and process more frames per second. Experiments with several datasets and event detection tasks on an edge node confirm that Mainstream improves mean event detection F1-scores by up to 47% relative to a static approach of retraining only the last DNN layer and sharing all others ("Max-Sharing") and by 87X relative to the common approach of using fully independent per-application DNNs ("No-Sharing").



Mainstream Architecture. Offline, for each task, M-Trainer takes a labeled dataset and outputs an M-Package. M-Scheduler takes independently generated M-Packages, and chooses the task-specific degree of specialization and frame rate. M-Scheduler deploys the unified multi-task model to M-Runner, performing inference on the edge.

Tributary: Spot-dancing for Elastic Services with Latency SLOs

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

2018 USENIX Annual Technical Conference. July 11–13, 2018 Boston, MA.

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

A Case for Packing and Indexing in Cloud File Systems

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

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

Small (kilobyte-sized) objects are the bane of highly scalable cloud object stores. Larger (at least megabyte-sized) objects not only improve performance, but also result in orders of magnitude lower cost, due to the current operation-based pricing model of commodity cloud object stores. For example, in Amazon S3's current pricing scheme, uploading 1GiB data by issuing 4KiB PUT requests (at 0.0005 cents each) is approximately 57X more expensive than storing that same 1GiB for a month. To address this problem, we propose client-side packing of small immutable files into gigabyte-sized blobs with embedded indices to identify each file's location. Experiments with a packing implementation in Alluxio (an open-source distributed file system) illustrate the potential benefits, such as simultaneously increasing file creation throughput by up to 60X and decreasing cost to 1/25000 of the original.

On the Diversity of Cluster Workloads and its Impact on Research Results

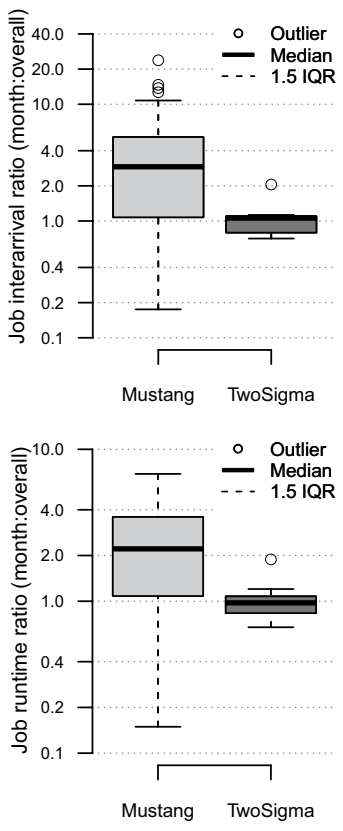
George Amvrosiadis, Jun Woo Park, Gregory R. Ganger, Garth A. Gibson, Elisabeth Baseman & Nathan DeBardeleben

2018 USENIX ATC '18, Boston, MA, July 11-13, 2018.

continued on page 26

continued from page 25

Six years ago, Google released an invaluable set of scheduler logs which has already been used in more than 450 publications. We find that the scarcity of other data sources, however, is leading researchers to overfit their work to Google's dataset characteristics. We demonstrate this overfitting by introducing four new traces from two private and two High Performance Computing (HPC) clusters. Our analysis shows that the private cluster workloads, consisting of data analytics jobs expected to be more closely related to the Google workload, display more similarity to the HPC cluster workloads. This observation suggests that additional traces should be considered when evaluating the generality of new research.



Is a month representative of the overall workload? The boxplots show distributions of the average job inter-arrival period (left) and duration (right) per month, normalized by the trace's overall average. Boxplot whiskers are defined at 1.5 times the distribution's Inter-Quartile Range (standard Tukey boxplots).

To aid the community in moving forward, we release the four analyzed traces, including: the longest publicly available trace spanning all 61 months of an HPC cluster's lifetime and a trace from a 300,000-core HPC cluster, the largest cluster with a publicly available trace. We present an analysis of the private and HPC cluster traces that spans job characteristics, workload heterogeneity, resource utilization, and failure rates. We contrast our findings with the Google trace characteristics and identify affected work in the literature. Finally, we demonstrate the importance of dataset plurality and diversity by evaluating the performance of a job runtime predictor using all four of our traces and the Google trace.

Mosaic: Enabling Application-Transparent Support for Multiple Page Sizes in Throughput Processors

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

ACM SIGOPS Operating System Review - Special Topics, Vol. 52, Issue 1, July 2018

Contemporary discrete GPUs support rich memory management features such as virtual memory and demand paging. These features simplify GPU programming by providing a virtual address space abstraction similar to CPUs and eliminating manual memory management, but they introduce high performance overheads during (1) address translation and (2) page faults. A GPU relies on high degrees of thread-level parallelism (TLP) to hide memory latency. Address translation can undermine TLP, as a single miss in the translation lookaside buffer (TLB) invokes an expensive serialized page table walk that often stalls multiple threads. Demand paging can also undermine TLP, as multiple threads often stall while they wait for an expensive data transfer over the system I/O (e.g., PCIe) bus when the GPU demands a page. In modern GPUs, we face a trade-off on how the page size used for memory management affects address translation and demand paging. The address translation overhead is lower when we employ a larger page size (e.g., 2MB large pages, compared with conventional 4KB base pages), which increases TLB coverage and thus reduces TLB misses. Conversely, the demand paging overhead is lower when we employ a smaller page size, which decreases the system I/O bus transfer latency. Support for multiple page sizes can help relax the page size trade-off so that address translation and demand paging optimizations work together synergistically. However, existing page coalescing (i.e., merging base pages into a large page) and splintering (i.e., splitting a large page into base pages) policies require costly base page migrations that undermine the benefits multiple page sizes provide. In this paper, we observe that GPGPU applications present an opportunity to support multiple page sizes without costly data migration, as the applications perform most of their memory allocation en masse (i.e., they allocate a large number of base pages at once). We show that this en masse allocation allows us to create intelligent memory allocation policies which ensure that base pages that are contiguous in virtual memory are allocated to contiguous physical memory pages. As a result, coalescing and splintering operations no longer need to migrate base pages.

We introduce Mosaic, a GPU memory manager that provides application-transparent support for multiple page sizes. Mosaic uses base pages to transfer data over the system I/O bus, and allocates physical memory in a way that (1) preserves base page contiguity and (2) ensures that a large page frame contains pages from only a single memory protection domain. We take

continued on page 27

continued from page 26

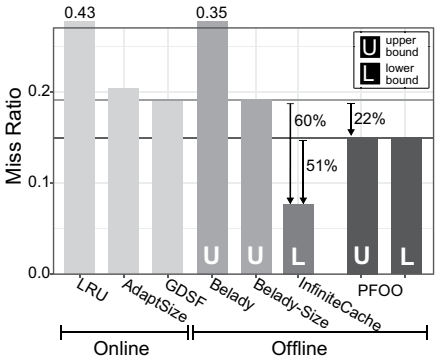
advantage of this allocation strategy to design a novel in-place page size selection mechanism that avoids data migration. This mechanism allows the TLB to use large pages, reducing address translation overhead. During data transfer, this mechanism enables the GPU to transfer only the base pages that are needed by the application over the system I/O bus, keeping demand paging overhead low. Our evaluations show that Mosaic reduces address translation overheads while efficiently achieving the benefits of demand paging, compared to a contemporary GPU that uses only a 4KB page size. Relative to a state-of-the-art GPU memory manager, Mosaic improves the performance of homogeneous and heterogeneous multi-application workloads by 55.5% and 29.7% on average, respectively, coming within 6.8% and 15.4% of the performance of an ideal TLB where all TLB requests are hits.

Practical Bounds on Offline Caching with Variable Object Sizes

Daniel Berger, Nathan Beckmann & Mor Harchol-Balter

Proc. ACM Meas. Anal. Comput. Syst., Vol. 2, No. 2, Article 32. June 2018. POMACS 2018.

Many recent caching systems aim to improve miss ratios, but there is no good sense among practitioners of how much further miss ratios can be improved. In other words, should the systems community continue working on this problem? Currently, there is no principled answer to this question. In practice, object sizes often vary by several orders of magnitude, where computing the optimal miss ratio (OPT) is known to be NP-hard. The few known results on caching with variable object sizes provide very weak bounds and are impractical to compute on traces of realistic length. We propose a new method to compute upper and lower bounds on OPT. Our key insight is to represent caching as a min-cost flow problem, hence we call our method the flow-based offline optimal (FOO). We prove that, under simple independence assumptions, FOO's bounds become tight as the number of objects goes to infinity. Indeed, FOO's error over 10M requests of production CDN and storage traces is negligible: at most 0.3%. FOO thus reveals, for the first time, the limits of caching with variable object sizes. While FOO is very accurate, it is computationally impractical on traces with hundreds of millions of requests. We therefore extend FOO to obtain more efficient bounds on OPT, which we call practical flow-based offline optimal (PFOO). We evaluate PFOO on several full production traces and use it to compare OPT to prior online policies. This analysis shows that current caching systems are in fact still far from optimal, suffering 11–43% more cache misses than OPT, whereas the best prior offline bounds suggest that there is essentially no room for improvement.



Miss ratios on a production CDN trace for a 4 GB cache. Prior to our work, the best prior upper bound on OPT is within 1% of online algorithms on this trace, leading to the false impression that there is no room for improvement. The only prior lower bound on OPT (Infinite-Cap) is 60% lower than the best upper bound on OPT (Belady-Size). By contrast, PFOO provides nearly tight upper and lower bounds, which are 22% below the online algorithms. PFOO thus shows that there is actually significant room for improving current caching algorithms.

Learning a Code: Machine Learning for Approximate Non-Linear Coded Computation

Jack Kosaian, K.V. Rashmi & Shivaram Venkataraman

arXiv:1806.01259v1 [cs.LG] 4 June, 2018.

Machine learning algorithms are typically run on large scale, distributed compute infrastructure that routinely face a number of unavailabilities such as failures and temporary slowdowns. Adding redundant computations using coding-theoretic tools called "codes" is an emerging technique to alleviate the adverse effects of such unavailabilities. A code consists of an encoding function that proactively introduces redundant computation and a decoding function that reconstructs unavailable outputs using the available ones. Past work focuses on using codes to provide resilience for linear computations and specific iterative optimization algorithms. However, computations performed for a variety of applications including inference on state-of-the-art machine learning algorithms, such as neural networks, typically fall outside this realm. In this paper, we propose taking a learning-based approach to designing codes that can handle non-linear computations. We present carefully designed neural network architectures and a training methodology for learning encoding and decoding functions that produce approximate reconstructions of unavailable computation results. We present extensive experimental results demonstrating the effectiveness of the proposed approach: we show that the our learned codes can accurately reconstruct 64 ?? 98% of the unavailable predictions from neural-network based image classifiers (multi-layer perceptron and ResNet-18) on the MNIST, Fashion-MNIST, and CIFAR-10 datasets. To the best of our knowledge, this work

continued on page 28

continued from page 27

proposes the first learning-based approach for designing codes, and also presents the first coding-theoretic solution that can provide resilience for any non-linear (differentiable) computation. Our results show that learning can be an effective technique for designing codes, and that learned codes are a highly promising approach for bringing the benefits of coding to non-linear computations.

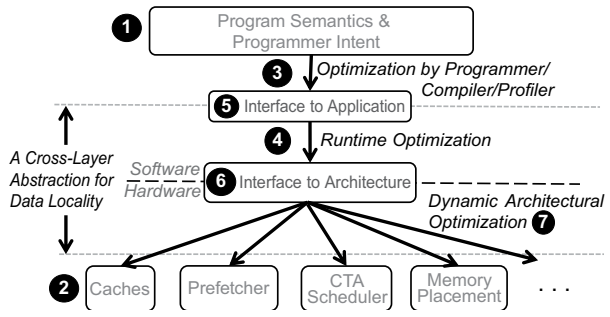
The Locality Descriptor: A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs

Nandita Vijaykumar, Eiman Ebrahimi, Kevin Hsieh, Phillip B. Gibbons & Onur Mutlu

The 45th International Symposium on Computer Architecture - June 2-6, ISCA 2018. Los Angeles, California, USA.

Exploiting data locality in GPUs is critical to making more efficient use of the existing caches and the NUMA-based memory hierarchy expected in future GPUs. While modern GPU programming models are designed to explicitly express parallelism, there is no clear explicit way to express data locality—i.e., reuse-based locality to make efficient use of the caches, or NUMA locality to efficiently utilize a NUMA system. On the one hand, this lack of expressiveness makes it a very challenging task for the programmer to write code to get the best performance out of the memory hierarchy. On the other hand, hardware-only architectural techniques are often suboptimal as they miss key higher-level program semantics that are essential to effectively exploit data locality.

In this work, we propose the Locality Descriptor, a crosslayer abstraction to explicitly express and exploit data locality in GPUs. The Locality Descriptor (i) provides the software a flexible and portable interface to optimize for data locality, requiring no knowledge of the underlying memory techniques and resources, and (ii) enables the architecture to leverage key program semantics and effectively coordinate a range of techniques (e.g., CTA scheduling, cache management, memory placement) to exploit locality.



Overview of the proposed holistic cross-layer abstraction. The goal is to connect program semantics and programmer intent (1) with the underlying architectural mechanisms (2). By doing so, we enable optimization at different levels of the stack: (i) as an additional knob for static code tuning by the programmer, compiler, or autotuner (3), (ii) runtime software optimization (4), and (iii) dynamic architectural optimization (7) using a combination of architectural techniques. This abstraction interfaces with a parallel GPU programming model like CUDA (5) and conveys key program semantics to the architecture through low overhead interfaces (6).

Query-based Workload Forecasting for Self-Driving Database Management Systems

Lin Ma, Dana Van Aken, Ahmed Hefny, Gustavo Mezerhane, Andrew Pavlo & Geoffrey J. Gordon

SIGMOD/PODS '18 International Conference on Management of Data Houston, TX, USA, June 10-15, 2018.

The first step towards an autonomous database management system (DBMS) is the ability to model the target application's workload. This is necessary to allow the system to anticipate future workload needs and select the proper optimizations in a timely manner. Previous forecasting techniques model the resource utilization of the queries. Such metrics, however, change whenever the physical design of the database and the hardware resources change, thereby rendering previous forecasting models useless.

We present a robust forecasting framework called QueryBot 5000 that allows a DBMS to predict the expected arrival rate of queries in the future based on historical data. To better support highly dynamic environments, our approach uses the logical composition of queries in the workload rather than the amount of physical resources used for query execution. It provides multiple horizons (short- vs. long-term) with different aggregation intervals. We also present a clustering-based technique for reducing the total number of forecasting models to maintain. To evaluate our approach, we compare our forecasting models against other state-of-the-art models on three real-world database traces. We implemented our models in an external controller for PostgreSQL and MySQL and demonstrate their effectiveness in selecting indexes.

Building a Bw-Tree Takes More Than Just Buzz Words

Ziqi Wang, Andrew Pavlo, Hyeontaek Lim, Viktor Leis, Huanchen Zhang, Michael Kaminsky & David G. Andersen

SIGMOD/PODS '18 International Conference on Management of Data Houston, TX, USA—June 10-15, 2018.

In 2013, Microsoft Research proposed the Bw-Tree (humorously termed the "Buzz Word Tree"), a lock-free index that provides high throughput for transactional database workloads in SQL Server's Hekaton engine. The Bw-Tree avoids locks by appending delta record to tree nodes and using an indirection layer that allows it to atomically update physical pointers using compare-and-

continued on page 29

continued from page 28

swap (CaS). Correctly implementing this techniques requires careful attention to detail. Unfortunately, the Bw-Tree papers from Microsoft are missing important details and the source code has not been released.

This paper has two contributions: First, it is the missing guide for how to build a lock-free Bw-Tree. We clarify missing points in Microsoft's original design documents and then present techniques to improve the index's performance. Although our focus here is on the Bw-Tree, many of our methods apply more broadly to designing and implementing future lock-free in-memory data structures. Our experimental evaluation shows that our optimized variant achieves 1.1–2.5× better performance than the original Microsoft proposal for highly concurrent workloads. Second, our evaluation shows that despite our improvements, the Bw-Tree still does not perform as well as other concurrent data structures that use locks.

A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory

Nandita Vijaykumar, Abhilasha Jain, Diptesh Majumdar, Kevin Hsieh, Gennady Pekhimenko, Eiman Ebrahimi, Nastaran Hajinazar, Phillip B. Gibbons & Onur Mutlu

45th International Symposium on Computer Architecture (ISCA), Los Angeles, CA, USA, June 2018.

This paper makes a case for a new cross-layer interface, Expressive Memory (XMem), to communicate higher-level program semantics from the application to the system software and hardware architecture. XMem provides (i) a flexible and extensible abstraction, called an Atom, enabling the application to express key program semantics in terms of how the program accesses data and the attributes of the data itself, and (ii) new cross-layer interfaces to make the expressed higher-level information available to the underlying OS and architecture.

Weiwei Gong (Oracle) talks about "Oracle Database In-Memory: Accelerating Joins and Aggregations" during the PDL Consortium Speakers Series at visit day 2018.

By providing key information that is otherwise unavailable, XMem exposes a new, rich view of the program data to the OS and the different architectural components that optimize memory system performance (e.g., caches, memory controllers).

By bridging the semantic gap between the application and the underlying memory resources, XMem provides two key benefits. First, it enables architectural/system-level techniques to leverage key program semantics that are challenging to predict or infer. Second, it improves the efficacy and portability of software optimizations by alleviating the need to tune code for specific hardware resources (e.g., cache space). While XMem is designed to enhance and enable a wide range of memory optimizations, we demonstrate the benefits of XMem using two use cases: (i) improving the performance portability of software-based cache optimization by expressing the semantics of data locality in the optimization and (ii) improving the performance of OS-based page placement in DRAM by leveraging the semantics of data structures and their access properties.

Better Caching in Search Advertising Systems with Rapid Refresh Predictions

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

Proceedings of the 2018 World Wide Web Conference on World Wide Web.

To maximize profit and connect users to relevant products and services, search advertising systems use sophisticated machine learning algorithms to estimate the revenue expectations of thousands of matching ad listings per query. These machine learning computations constitute a substantial part of the operating cost, e.g., 10% to 30% of the total gross revenues. It is desirable to cache and reuse previous computation results to reduce this cost, but caching introduces approximation which comes with potential revenue loss. To maximize cost savings while minimizing the overall revenue impact, an intelligent refresh policy is required to decide when to refresh the cached computation results. The state-of-the-art manually-tuned refresh heuristic uses revenue history to assign different refresh frequencies. Using the gradient boosting regression tree algorithm with well selected features, we introduce a rapid prediction framework that provides refresh decisions at higher accuracy compared to the heuristic. This enables us to build a prediction-based refresh policy and a cache achieving higher profit without manual parameter tuning. Simulations conducted on the logs from a major commercial search advertising system show that our proposed cache design reduces the negative revenue impact (0.07x), and improves the cost savings (1.41x) and the net profit (1.50~1.70x) compared to the state-of-the-art manually-tuned heuristic-based cache design.

Practical Bounds on Optimal Caching with Variable Object Sizes

Daniel S. Berger, Nathan Beckmann & Mor Harchol-Balter

ACM Proceedings on Measurement and Analysis of Computing Systems. Vol. 2, No. 2, Article 32. June 2018.

Many recent caching systems aim to improve miss ratios, but there is no

continued on page 30

continued from page 29

good sense among practitioners of how much further miss ratios can be improved. In other words, should the systems community continue working on this problem?

Currently, there is no principled answer to this question. In practice, object sizes often vary by several orders of magnitude, where computing the optimal miss ratio (OPT) is known to be NP-hard. The few known results on caching with variable object sizes provide very weak bounds and are impractical to compute on traces of realistic length.

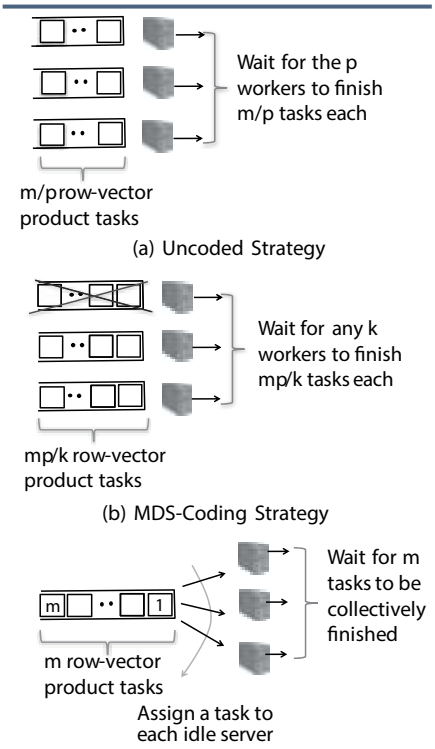
We propose a new method to compute upper and lower bounds on OPT. Our key insight is to represent caching as a min-cost flow problem, hence we call our method the flow-based offline optimal (FOO). We prove that, under simple independence assumptions, FOO's bounds become tight as the number of objects goes to infinity. Indeed, FOO's error over 10M requests of production CDN and storage traces is negligible: at most 0.3%. FOO thus reveals, for the first time, the limits of caching with variable object sizes. While FOO is very accurate, it is computationally impractical on traces with hundreds of millions of requests. We therefore extend FOO to obtain more efficient bounds on OPT, which we call practical flow-based offline optimal (PFOO).We evaluate PFOO on several full production traces and use it to compare OPT to prior online policies. This analysis shows that current caching systems are in fact still far from optimal, suffering 11–43% more cache misses than OPT, whereas the best prior offline bounds suggest that there is essentially no room for improvement.

Rateless Codes for Near-Perfect Load Balancing in Distributed Matrix-Vector Multiplication

Ankur Mallick, Malhar Chaudhari & Gauri Joshi

arXiv:1804.10331v2 [cs.DC] 30 Apr 2018.

Large-scale machine learning and data



In the uncoded scheme, we wait for the p workers to finish m/p row-vector products each. With MDS-coding, we only need to wait for k out of p workers, but each worker has to complete mp/k. The dynamic load-balancing strategy has a central queue of the m tasks, which are dynamically assigned to idle worker nodes.

mining applications require computer systems to perform massive computations that need to be parallelized across multiple nodes, for example, massive matrix-vector and matrix-matrix multiplication. The presence of straggling nodes – computing nodes that unpredictably slowdown or fail – is a major bottleneck in such distributed computations. We propose a rateless fountain coding strategy to alleviate the problem of stragglers in distributed matrix-vector multiplication. Our algorithm creates a stream of linear combinations of the m rows of the matrix, and assigns them to different worker nodes, which then perform row-vector products with the encoded rows. The original matrix-vector product can be decoded as soon as slightly more than m rowvector products are collectively finished by the nodes. This strategy enables fast nodes to steal work from slow nodes, without requiring the master to perform any dynamic load-balancing. Compared to recently proposed fixed-rate erasure coding strategies which ignore partial work done by straggling nodes, rateless coding achieves significantly lower overall delay, as well as small computational and decoding overhead.

Implicit Decomposition for Write-Efficient Connectivity Algorithms

Naama Ben-David, Guy E. Blelloch, Jeremy T. Fineman, Phillip B. Gibbons, Yan Gu, Charles McGuffey & Julian Shun

2018 International Parallel and Distributed Processing Symposium (IPDPS '18). May 21-25, 2018, Vancouver, BC, Canada.

The future of main memory appears to lie in the direction of new technologies that provide strong capacity-to-performance ratios, but have write operations that are much more expensive than reads in terms of latency, bandwidth, and energy. Motivated by this trend, we propose sequential and parallel algorithms to solve graph connectivity problems using significantly fewer writes than conventional algorithms. Our primary algorithmic tool is the construction of an o(n)-sized implicit decomposition of a bounded-degree graph G, which combined with read-only access to G enables fast answers to connectivity and biconnectivity queries on G. The construction breaks the linear-write "barrier", resulting in costs that are asymptotically lower than conventional algorithms while adding only a modest cost to querying time. For general non-sparse graphs, we also provide the first o(m) writes and O(m) operations parallel algorithms for connectivity and biconnectivity. These algorithms provide insight into how applications can efficiently process computations

continued on page 31

continued from page 30

on large graphs in systems with read-write asymmetry.

Intermittent Deep Neural Network Inference

Graham Gobieski, Nathan Beckmann & Brandon Lucia

SysML 2018, February 15-16, 2018, Stanford, CA.

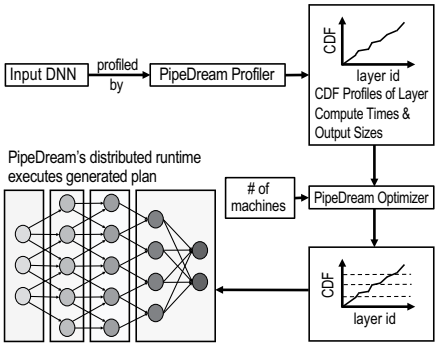
The maturation of energy-harvesting technology has enabled new classes of sophisticated, batteryless systems that will drive the next wave of Internet of Things (IoT) applications. These applications require intelligence at the edge and even in the sensor node, e.g., allowing systems to immediately interpret sensed data and make judicious use of scarce bandwidth. However, inference on energy-harvesting devices presents challenges currently unexplored, namely that energy-harvesting devices are severely resource-constrained and operate intermittently only when energy is available. Typical systems run at a few MHz, have a few hundred KBs of memory and consume power under 1mW when active [8, 22]. In comparison, the most energy efficient DNN inference accelerators consume hundreds of mW.

PipeDream: Fast and Efficient Pipeline Parallel DNN Training

Aaron Harlap, Deepak Narayanan, Amar Phanishayee, Vivek Seshadri, Nikhil Devanur, Greg Ganger & Phil Gibbons

SysML '18, Feb. 15-16, 2018.

Stanford, CA. PipeDream is a Deep Neural Network (DNN) training system for GPUs that parallelizes computation by pipelining execution across multiple machines. Its pipeline parallel computing model avoids the slowdowns faced by data-parallel training when large models and/or limited network bandwidth induce high communication-to-computation ratios. PipeDream reduces communication by up to 95% for large DNNs relative to data-parallel training, and allows perfect overlap of communication and computation. PipeDream keeps all available GPUs productive by systematically partitioning DNN layers among them to balance work and minimize communication, versions model parameters for backward pass correctness, and schedules the forward and backward passes of different inputs in round-robin fashion to optimize "time to target accuracy". Experiments with five different DNNs on two different clusters show that PipeDream is up to 5x faster in time-to-accuracy compared to data-parallel training.



PipeDream's automated mechanism to partition DNN layers into stages. PipeDream first profiles the input DNN, to get estimates for each layer's compute time and output size. Using these estimates, PipeDream's optimizer partitions layers across available machines.

YEAR IN REVIEW

continued from page 4

national Symposium on Computer Performance, Modeling, Measurements, and Evaluation (Performance 2018) in Toulouse, France and received the Best Student Paper Award!

- Daniel S. Berger presented "Towards Lightweight and Robust Machine Learning for CDN Caching" at HotNets-XVII, in Redmond, WA, USA.
- Qing Zheng presented "Scaling Embedded In-Situ Indexing with DeltaFS" at SC18, in Dallas, Texas, USA.

October 2018

- 26th annual PDL Retreat.
- Ben Blum successfully defended his Ph.D. research on "Practical Concurrency Testing or: How I Learned to Stop Worrying and Love the Exponential Explosion."
- Jinliang Wei presented his thesis proposal on "Efficient and Programmable Distributed Shared Memory Systems for Machine Learning Training."
- Kevin Hsieh presented "Focus: Querying Large Video Datasets with Low Latency and Low Cost" at the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI), in Carlsbad, CA.
- Ziv Scully presented "SOAP Bubbles: Robust Scheduling Under Adversarial Noise" at the 56th Annual Allerton Conference on Communication, Control, and Computing, in Monticello, IL.
- Anurag Mukkara presented "Exploiting Locality in Graph Analytics through Hardware-Accelerated Traversal Scheduling" at the 51st Annual IEEE/ACM International Symposium on Microarchitecture in Fukuoka, Japan.
- Daniel Berger presented "Robin-

continued on page 32

continued from page 31

Hood: Tail Latency Aware Caching—Dynamic Reallocation from Cache-Rich to Cache-Poor" at OSDI '18 in Carlsbad, CA, USA.

- Andrew Chung and Jun Woo Park presented the Best Student Paper to SoCC '18 titled "Stratus: Cost-aware Container Scheduling in the Public Cloud."

September 2018

- Jin Kyu Kim successfully defended his Ph.D. dissertation on the "Framework Design for Improving Computational Efficiency and Programming Productivity for Distributed Machine Learning."
- PDL Alum Wei Dai was a winner of the Pittsburgh Business Times 30 under 30 Award!

August 2018

- Nandita Vijaykumar proposed her Ph.D. research "Rethinking Cross-layer Abstractions to Enhance Programmability, Portability, and Performance."
- Kevin Hsieh proposed his thesis research on "Low-Latency, Low-Cost Machine Learning Systems on Large-Scale, Highly-Distributed Data."

July 2018

- Joy James Prabhu Arulraj successfully defended his Ph.D. dissertation on "The Design and Implementation of a Non-Volatile Memory Database Management System."
- Saurabh Kadekodi presented "Geriatrix: Aging what you see and what you don't see—A file system aging approach for modern storage systems" at the 2018 USENIX Annual Technical Conference in Boston, MA.
- Sol Boucher presented "Putting the 'Micro' Back in Microservice" at the 2018 USENIX Annual Technical Conference in Boston, MA.
- Aaron Harlap presented "Tributary: Spot-dancing for Elastic Services with Latency SLOs" at the 2018 USENIX Annual Technical Conference in Boston, MA.
- George Amvrosiadis presented "On the Diversity of Cluster Workloads and its Impact on Research Results" at the 2018 USENIX Annual Technical Conference in Boston, MA.
- Angela Jiang presented "Mainstream: Dynamic Stem-Sharing for Multi-Tenant Video Processing," at the 2018 USENIX Annual Technical Conference in Boston, MA.
- Saurabh Kadekodi presented "A Case for Packing and Indexing in Cloud File Systems" at the 10th USENIX Workshop on Hot Topics in Cloud Computing in Boston, MA.

June 2018

- Nandita Vijaykumar presented "The Locality Descriptor: A Holistic Cross-Layer Abstraction to Express Data Locality in GPUs" at the 45th International Symposium on Computer Architecture in Los Angeles, CA.
- Lin Ma presented "Query-based Workload Forecasting for Self-Driving Database Management Systems" at SIGMOD/PODS '18 in Houston, TX.
- Lin Ma presented "A Case for Richer Cross-layer Abstractions: Bridging the Semantic Gap with Expressive Memory" at ISCA'18 in Los Angeles, CA.
- Conglong Li presented "Better Caching in Search Advertising Systems with Rapid Refresh Predictions" at the 2018 Conference on the World Wide Web.
- Qing Zheng and Michael Kuchnik interned with LANL during the summer of 2018.
- Brian Schwedock interned with Google NYC as a Software Engineering Intern.

May 2018

- Jun Woo Park proposed his thesis research on "Distribution-based Cluster Scheduling."
- Aaron Harlap proposed his thesis research on "Improving ML Applications in Shared Computing Environments."
- SuRF: Practical Range Query Filtering with Fast Succinct Tries, presented by Huanchen Zhang at 2018 SIGMOD won Best Paper Award!
- Tianyu Li submitted his Master's thesis on "Supporting Hybrid Workloads for In-Memory Database Management Systems via a Universal Columnar Storage Format."
- 20th annual PDL Spring Visit Day.

Nandita Vijaykumar, PDL Ph.D. Graduate Student answers questions about her talk on "Expressive Memory Rethinking the Hardware-Software Contract with Rich Cross-Layer Abstractions" at the 2018 PDL Retreat.