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

Broadcom
Citadel
EMC Corporation
Facebook
Google
Hewlett-Packard Labs
Hitachi, Ltd.
Intel Corporation
Microsoft Research
MongoDB
NetApp, Inc.
Oracle Corporation
Samsung Information Systems America
Seagate Technology
Tintri
Two Sigma
Uber
Veritas
Western Digital

PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • SPRING 2016
http://www.pdl.cmu.edu/

Big-learning Systems for Big Data

Compiled by Greg Ganger for the PDL Big-learning Group

Data analytics (a.k.a. Big Data) has emerged as a primary data processing activity for business, science, and online services that attempt to extract insight from quantities of observation data. Increasingly, such analytics center on statistical machine learning (ML), in which an algorithm determines model parameters that make a chosen statistical model best fit the training data. Once fit (trained), such models can expose relationships among data items (e.g., for grouping documents into topics), predict outcomes for new data items based on selected characteristics (e.g., for recommendation systems), correlate effects with causes (e.g., for genomic analyses of diseases), and so on.

Growth in data sizes and desired model precision generally dictates parallel execution of ML algorithms on clusters of servers. Naturally, parallel ML involves the same work distribution, synchronization and data consistency challenges as other parallel computations. The PDL big-learning group has attacked these challenges, creating and refining powerful new approaches for supporting large-scale ML on Big Data. This short article overviews an inter-related collection of our efforts in this space.

Iterative Convergent ML

Most modern ML approaches rely on *iterative convergent algorithms*, such as stochastic gradient descent (SGD), to determine model parameter values. These algorithms start with some guess at a solution (a set of parameter values) and refine this guess over a number of iterations over the training data, improving an explicit goodness-of-solution objective function until sufficient convergence or goodness has been reached. Generally speaking, parallel realizations of iterative ML partition training data among the worker threads (running on available cores of each server used) that each contribute to computing the derived parameter values.

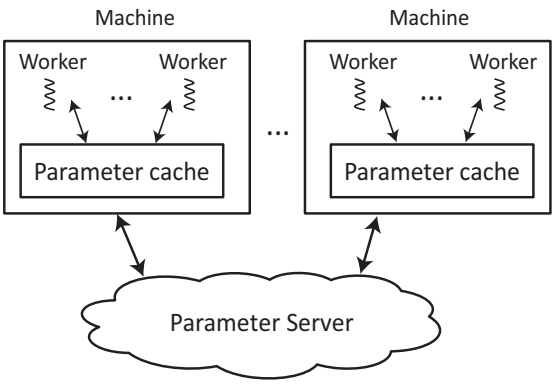


Figure 1: Parallel ML with parameter server.

Historically, the most common parallel execution model for iterative ML has been based on the Bulk Synchronous Parallel (BSP) model. In BSP, each thread executes a given amount of work on a private copy of shared state and barrier synchronizes with the others. Once all threads reach the barrier, updates are exchanged

continued on page 12

Greg Ganger

Hello from fabulous Pittsburgh!

It has been another great year for the Parallel Data Lab. Some highlights include several "best paper" awards for PDL publications, awards and fellowships for PDL students, and awards for PDL faculty. PDL's database systems research continues to build up strongly, research in cloud computing and "Big Data" systems keeps growing in scope, and PDL's storage systems and cloud classes evolve and grow ever more popular. Along the way, many students graduated and joined PDL Consortium companies, new students joined the PDL, and many cool papers have been published. Let me highlight a few things.

I'm excited by the resurgence of PDL database systems research induced by Andy's great energy and ideas. Cool results and papers are being produced on automated database tuning, deduplication in databases, efficient DB memory usage, and better exploitation of NVM in databases. The open source Peloton DBMS that his group is building combines several of these activities into what he refers to as a "self-driving" database, seeking to achieve autonomous adaptation to workload and resource conditions. It reminds me of PDL's previous efforts on Self-* Storage, but in a DBMS context and with powerful new analytical tools now available from the ML and queuing theory communities. Very cool.

I'm also excited about the growth and evolution of the storage and cloud classes created and led by PDL faculty. We introduced a new project in the storage systems class, replacing the long-used myFSCK project: the students develop an FTL (myFTL) for a NAND Flash SSD. We supply them with an SSD simulator that includes basic interfaces for read-page, write-page, and erase-block. The students write FTL code to maintain LBA-to-physical mappings, perform cleaning, and address wear leveling. As per usual, this first offering of the project was bumpy, but we're already working hard on refining it for this Fall. We expect ~150 MS students in the storage class, and our EC2-project-intensive cloud computing class is similarly popular. In addition to our lectures and the projects, the storage class features 4-6 corporate guest lecturers (thank you, PDL Consortium members!) bringing insight on real-world storage, trends, and futures.

I've said it before, but we continue to be energized by working at the core of two of today's largest growth areas: cloud computing and Big Data. For example, our explorations into new systems designs for supporting large-scale machine learning (a primary component of Big Data analytics) continues to provide great opportunities for impact via collaboration with Carnegie Mellon's excellent machine learning faculty. While map-reduce-style data processing, via frameworks like Hadoop and Spark, are great for fairly simple data processing, advanced machine learning requires different approaches to achieve high efficiency. Most ML-focused frameworks use an architecture based on parameter servers, which is a specialized key-value store. Of course, PDL has a great track record with key-value store research, and we've discovered a lot of powerful ways of specializing for ML algorithm properties. See the front-page article for more.

The breadth of analytics frameworks and other cloud computing activities leads to resource scheduling challenges. Our TetriSched project is developing new ways of allowing users to express their per-job resource type preferences (e.g., machine locality or hardware accelerators) and then exploring the trade-offs among them to maximize utility of the public and/or private cloud infrastructure. Our most recent publication on this work just received PDL's most recent research award, being named Best Student Paper at Eurosys 2016 in April. We are also exploring

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.

FACULTY

Greg Ganger (pdl director)
412•268•1297
ganger@ece.cmu.edu

David Andersen
Lujo Bauer
Chuck Cranor
Lorrie Cranor
Christos Faloutsos
Kayvon Fatahalian
Eugene Fink
Rajeev Gandhi
Phil Gibbons
Garth Gibson
Seth Copen Goldstein
Mor Harchol-Balter
Todd Mowry
Onur Mutlu
Priya Narasimhan
David O'Hallaron
Andy Pavlo
Majd Sakr
M. Satyanarayanan
Srinivasan Seshan
Alex Smola
Hui Zhang

STAFF MEMBERS

Bill Courtright, 412•268•5485
(pdl executive director) wcourtright@cmu.edu
Karen Lindenfelser, 412•268•6716
(pdl administrative manager) karen@ece.cmu.edu
Jason Boles
Joan Digney
Chad Dougherty
Mitch Franzos
Charlene Zang

VISITING RESEARCHERS / POST DOCS

Saugata Ghose
Atsushi Kawamura
Hyeontaek Lim
Rolando Martins
Raja Sambasivan
Michael (Tieying) Zhang

GRADUATE STUDENTS

Abutalib Aghayev
Joy Arulraj
Rachata Ausavarungnirun
Ben Blum
Amirali Boroumand
Sol Boucher
Lei Cao
Kevin Chang
Henggang Cui
Wei Dai
Debanshu Das
Utsav Drolia
Jian Fang
Kristen Scholes Gardner
Kiryong Ha
Aaron Harlap
Kevin Hsieh
Angela Jiang
Junchen Jiang
Wesley Jin
Saurabh Arun Kadekodi
Anuj Kalia
Mike Kasick
Rajat Kateja
Jin Kyu Kim
Thomas Kim
Dimitris Konomis
Conglong Li
Mu Li
Yang Li
Yixin Luo
Lin Ma
Prashanth Menon
Nathan Mickulicz
Ravi Teja Mullapudi
Jun Woo Park
Gennady Pekhimenko
Matt Perron
Alex Poms
Aurick Qiao
Kai Ren
Vivek Seshadri
Aditya Shantanu
Jiaqi Tan
Alexey Tumanov
Dana Van Aken
Nandita Vijaykumar
Jinliang Wei
Lin Xiao
Hongyi Xin
Lianghong Xu
Jiajun Yao
Huanchen Zhang
Rui Zhang
Qing Zheng
Dong Zhou
Timothy Zhu

FROM THE DIRECTOR'S CHAIR

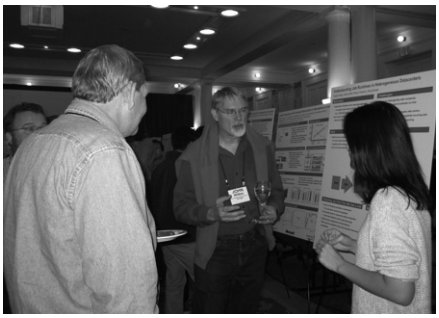
interesting ways of having ML frameworks collaborate with the scheduler, such as in having a parameter server system exploit the Amazon EC2 spot-price market to elastically grow and shrink.

Our explorations of how systems should be designed differently to exploit and work with new underlying storage technologies, such as NVM and Flash SSDs, continues to expand on many fronts. Activities range from new search and sort algorithms that understand the asymmetry between read and write performance (i.e., assuming using NVM directly) to FTL interface/implementation changes to reduce read tail latencies. There is a major push to design new database and index approaches for NVM and hybrid memory systems. We're excited about continuing to work with PDL companies on understanding where the hardware is (and should be) going and how it should be exploited in systems.

Naturally, PDL's long-standing focus on scalable storage continues strongly. As always, a primary challenge is metadata scaling, and PDL researchers are exploring several approaches to dealing with scale along different dimensions. For example, Garth's students are combining our recent IndexFS approach with novel sharding approaches to scale both within and across server nodes. A cool new direction being explored is allowing high-end approaches to manage their own namespaces, bypassing traditional metadata bottlenecks entirely.

Many other ongoing PDL projects are also producing cool results. For example, our key-value store research continues to expose new approaches to achieving faster and more efficient systems. Our "caveat scriptor" approach for Shingled magnetic recording (SMR) disks involves letting the host software be responsible for managing data placement rather than hiding it in an FTL-like layer inside the drive, with the promise of greater efficiency from integrating with OS or DBMS policies. Our continued operation of private clouds in the Data Center Observatory (DCO) serves the dual purposes of providing resources for real users (CMU researchers) and providing us with invaluable Hadoop logs, instrumentation data, and case studies. 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.



Angela Jiang describes her research on "Understanding Job Runtimes in Heterogeneous Datacenters" to Jerry Fredin (NetApp) and John Wilkes (Google) at the 2015 PDL Retreat.

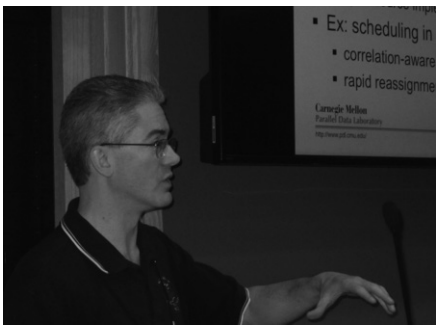
Greg poses with Bianca Schroeder, Associate Professor and Canada Research Chair in the Computer Science Department at the University of Toronto, on the occasion of her being named a PDL Distinguished Alumni. Bianca was a member of the PDL from 1999 – 2007.

May 2016

- 18th annual PDL Spring Visit Day.
- Rajat Kateja will be interning at Microsoft Research Redmond with Anirudh Badam this summer.
- Huanchen Zhang is going to be interning at HP Labs this summer with Kim Keeton.

April 2016

- Joy Arulraj received a Samsung PhD fellowship.
- Alexey Tumanov and his co-authors won Best Student Paper at Eurosys'16 in London, UK, for their work on "TetriSched: Global Rescheduling with Adaptive Plan-ahead in Dynamic Heterogeneous Clusters"
- Also presented at Eurosys: "GeePS: Scalable Deep Learning on Distributed GPUs with a GPU-Specialized Parameter Server" (Henggang Cui) and "STRADS: A Distributed Framework for Scheduled Model Parallel Machine Learning" (Jin Kyu Kim)
- Rajat Kateja won Best Graduate Poster for his presentation of "TailTrimmer: Reducing Read Tail Latencies in SSDs" at the 2016 Industry Academia Partnership CMU Cloud Workshop.
- Dana Van Aken won a 2016 National Science Foundation graduate fellowship.
- Lianghong Xu successfully defended his PhD research on "Similarity-based Deduplication for Databases."
- Mor Harchol-Balter gave several keynote talks and distinguished lectures throughout the year, speaking at CanQueue 2016, ACM SIGMETRICS 2016, IMaCCS 2016 and ICDCS 2015.
- Joy Arulraj gave his speaking skills talk on "Peloton: Street Strength Database Management System for Real-Time Analytics."
- Junchen Jiang gave his speaking skills talk on "CFA: A Practical Prediction System for Video QoE Optimization."



Greg gives an overview of PDL research to our industry visitors during the opening session of the 2015 PDL Retreat.

March 2016

- Vivek Seshadri successfully defended his PhD research on "Simple DRAM and Virtual Memory Abstractions to Enable Highly Efficient Memory Subsystems."
- Three papers were presented at the 22nd HPCA in Barcelona, Spain: "SizeCap: Efficiently Handling Power Surges in Fuel Cell Powered Data Centers" (Yang Li), "Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM (Kevin Chang), and "A Case for Toggle-Aware Compression for GPU Systems" (Gennady Pekhimenko).

February 2016

- Kristy Gardner received the SCS Graduate Student Teaching Award.
- Hyeontaek Lim presented "Towards Accurate and Fast Evaluation of Multi-Stage Log-Structured Designs" at FAST'16 in Santa Clara, CA.

January 2016

- Donghyuk Lee presented "Simultaneous Multi-Layer Access: Improving 3D-Stacked Memory Bandwidth at Low Cost" at HiPEAC'16 in Prague, Czech Republic.

December 2015

- Lorrie Cranor was appointed as the Federal Trade Commission's Chief Technologist.
- Onur Mutlu, gave a keynote talk on "Rethinking Memory System Design (along with Interconnects)" at the 8th International Workshop on Network on Chip Architectures (NoCArc), Honolulu, HI.
- Michael P. Kasick successfully defended his PhD research on "Black-Box Problem Diagnosis in Parallel File Systems."
- Yoshihisa Abe successfully defended his PhD dissertation on "Liberating Virtual Machines from Physical Boundaries through Execution Knowledge."
- Kristen Gardner proposed her PhD research on "Analyzing Systems with Redundant Requests."
- Jiaqi Tan proposed his PhD research on "Provable, Programmer-Visible Control-Flow Integrity for Software."

November 2015

- Timothy Zhu proposed his PhD thesis "Meeting Tail Latency SLOs in Shared Networked Storage."
- Jin Kyu Kim presented his speaking skills talk on "STRADS: Parallelizing ML over Model Parameters."
- Qing Zheng presented "DeltaFS: Exascale File Systems Scale Better Without Dedicated Servers" at PDSW'15 in Austin, TX.

October 2015

- 23rd annual PDL Retreat.

September 2015

- Jiaqi Tan won Best Contribution Award at the Student Forum for Formal Methods in Computer-Aided Design.

continued on page 28

GeePS: Scalable Deep Learning on Distributed GPUs with a GPU-Specialized Parameter Server

Henggang Cui, Hao Zhang, Gregory R. Ganger, Phillip B. Gibbons & Eric P. Xing.

ACM European Conference on Computer Systems, 2016 (EuroSys'16), 18th-21st April, 2016, London, UK.

Large-scale deep learning requires huge computational resources to train a multi-layer neural network. Recent systems propose using 100s to 1000s of machines to train networks with tens of layers and billions of connections. While the computation involved can be done more efficiently on GPUs than on more traditional CPU cores, training such networks on a single GPU is too slow and training on distributed GPUs can be inefficient, due to data movement overheads, GPU stalls, and limited GPU memory. This paper describes a new parameter server, called GeePS, that supports scalable deep learning across GPUs distributed among multiple machines, overcoming these obstacles. We show that GeePS enables a state-of-the-art single-node GPU implementation to scale well, such as to 13 times the number of training images processed per second on 16 machines (relative to the original optimized single-node code). Moreover, GeePS achieves a higher training throughput with just four GPU machines than that a state-of-the-art CPU-only system achieves with 108 machines.

TetriSched: Global Rescheduling with Adaptive Plan-ahead in Dynamic Heterogeneous Clusters

Alexey Tumanov, Timothy Zhu, Jun Woo Park, Michael A. Kozuch, Mor Harchol-Balter & Gregory R. Ganger

ACM European Conference on Computer Systems, 2016 (EuroSys'16), 18th-21st April, 2016, London, UK.

TetriSched is a scheduler that works in tandem with a calendaring reservation system to continuously re-evaluate the immediate-term scheduling plan for all pending jobs (including those with reservations and best-effort jobs) on each scheduling cycle. TetriSched leverages information supplied by the reservation system about jobs' deadlines and estimated runtimes to plan ahead in deciding whether to wait for a busy preferred resource type (e.g., machine with a GPU) or fall back to less preferred placement options. Plan-ahead affords significant flexibility in handling mis-estimates in job runtimes specified at reservation time. Integrated with the main reservation system in Hadoop YARN, TetriSched is experimentally shown to achieve significantly higher SLO attainment and cluster utilization than the best-configured YARN reservation and CapacityScheduler stack deployed on a real 256 node cluster.

Similarity-based Deduplication for Databases

Lianghong Xu, Andrew Pavlo, Sudipta Sengupta & Gregory R. Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-16-101, April 2016.

dbDedup is a similarity-based deduplication scheme for on-line database management systems (DBMSs). Beyond block-level compression of individual database pages or operation log (oplog) messages, as used in today's DBMSs, dbDedup uses byte-level delta encoding of individual records within the database to achieve greater savings. dbDedup's single-pass encoding method can be integrated into the storage and logging components of a DBMS to provide two benefits: (1) reduced size of data stored on disk beyond what traditional compression schemes provide, and (2) reduced amount of data transmitted over the network for replication services. To evaluate our work, we implemented dbDedup in a distributed NoSQL DBMS and analyzed its properties using four real datasets. Our results show that dbDedup achieves up to 37 reduction in the storage size and replication traffic of the database on its own and up to 61 reduction when paired with the DBMS's blocklevel compression. dbDedup provides both benefits with negligible effect on DBMS throughput or client latency (average and tail).

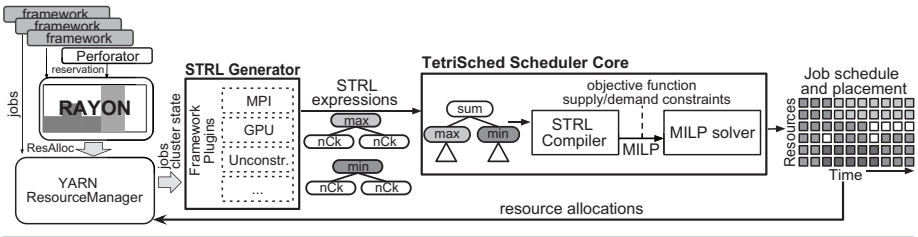
STRADS: A Distributed Framework for Scheduled Model Parallel Machine Learning

Jin Kyu Kim, Qirong Ho, Seunghak Lee, Xun Zheng, Wei Dai, Garth A. Gibson & Eric P. Xing

ACM European Conference on Computer Systems, 2016 (EuroSys'16), 18th-21st April, 2016, London, UK.

Machine learning (ML) algorithms are commonly applied to big data, using distributed systems that partition the data across machines and allow each machine to read and update all ML model parameters — a strategy known as data parallelism. An alternative and complimentary strategy, model parallelism, partitions the model parameters for non-shared parallel access

continued on page 6



TetriSched system architecture.

continued from page 5

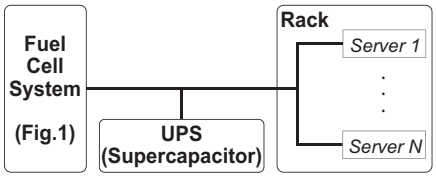
and updates, and may periodically repartition the parameters to facilitate communication. Model parallelism is motivated by two challenges that data-parallelism does not usually address: (1) parameters may be dependent, thus naive concurrent updates can introduce errors that slow convergence or even cause algorithm failure; (2) model parameters converge at different rates, thus a small subset of parameters can bottleneck ML algorithm completion. We propose scheduled model parallelism (SchMP), a programming approach that improves ML algorithm convergence speed by efficiently scheduling parameter updates, taking into account parameter dependencies and uneven convergence. To support SchMP at scale, we develop a distributed framework STRADS which optimizes the throughput of SchMP programs, and benchmark four common ML applications written as SchMP programs: LDA topic modeling, matrix factorization, sparse least-squares (Lasso) regression and sparse logistic regression. By improving ML progress per iteration through SchMP programming whilst improving iteration throughput through STRADS we show that SchMP programs running on STRADS outperform non model-parallel ML implementations: for example, SchMP LDA and SchMP Lasso respectively achieve 10x and 5x faster convergence than recent, well-established baselines.

SizeCap: Efficiently Handling Power Surges in Fuel Cell Powered Data Centers

Yang Li, Di Wang, Saugata Ghose, Jie Liu, Sriram Govindan, Sean James, Eric Peterson, John Siegler, Rachata Ausavarungnirun & Onur Mutlu,

22nd International Symposium on High Performance Computer Architecture (HPCA), March 12-16, Barcelona, Spain, 2016.

Fuel cells are a promising power source for future data centers, offering high



Configuration of a fuel cell powered data center.

energy efficiency, low greenhouse gas emissions, and high reliability. However, due to mechanical limitations related to fuel delivery, fuel cells are slow to adjust to sudden increases in data center power demands, which can result in temporary power shortfalls. To mitigate the impact of power shortfalls, prior work has proposed to either perform power capping by throttling the servers, or to leverage energy storage devices (ESDs) that can temporarily provide enough power to make up for the shortfall while the fuel cells ramp up power generation. Both approaches have disadvantages: power capping conservatively limits server performance and can lead to service level agreement (SLA) violations, while ESD-only solutions must significantly overprovision the energy storage device capacity to tolerate the shortfalls caused by the worstcase (i.e., largest) power surges, which greatly increases the total cost of ownership (TCO).

We propose SizeCap, the first ESD sizing framework for fuel cell powered data centers, which coordinates ESD sizing with power capping to enable a cost-effective solution to power shortfalls in data centers. SizeCap sizes the ESD just large enough to cover the majority of power surges, but not the worst-case surges that occur infrequently, to greatly reduce TCO. It then uses the smaller capacity ESD in conjunction with power capping to cover the power shortfalls caused by the worst-case power surges. As part of our new flexible framework, we propose multiple power capping policies with different degrees of awareness of fuel cell and workload behavior, and evaluate their impact on workload performance and ESD size. Using traces from Microsoft's production data center systems, we demonstrate that SizeCap significantly reduces the ESD size (by 85% for a workload with infrequent yet large power surges, and by 50% for a workload with frequent power surges) without violating any SLAs.

Full-Stack Architecting to Achieve a Billion Requests Per Second Throughput on a Single Key-Value Store Server Platform

Sheng Li, Hyeontaek Lim, Victor Lee, Jung Ho Ahn, Anuj Kalia, Michael Kaminsky, David G. Andersen, Seongil O, Sukhan Lee & Pradeep Dubey

ACM Transactions on Computer Systems (TOCS), Vol. 34, No. 2, April 2016.

Distributed in-memory key-value stores (KVSs), such as memcached, have become a critical data serving layer in modern Internet-oriented data center infrastructure. Their performance and efficiency directly affect the QoS of web services and the efficiency of data centers. Traditionally, these systems have had significant overheads from inefficient network processing, OS kernel involvement, and concurrency control. Two recent research thrusts have focused on improving key-value performance. Hardware-centric research has started to explore specialized platforms including FPGAs for KVSs; results demonstrated an order of magnitude increase in throughput and energy efficiency over stock memcached. Software-centric research revisited the KVS application to address fundamental software bottlenecks and to exploit the full potential of modern commodity hardware; these efforts also showed orders of magnitude improvement over stock memcached.

We aim at architecting high-performance and efficient KVS platforms,

continued on page 7

continued from page 6

and start with a rigorous architectural characterization across system stacks over a collection of representative KVS implementations. Our detailed full-system characterization not only identifies the critical hardware/software ingredients for high-performance KVS systems but also leads to guided optimizations atop a recent design to achieve a record-setting throughput of 120 million requests per second (MRPS) (167MRPS with client-side batching) on a single commodity server. Our system delivers the best performance and energy efficiency (RPS/watt) demonstrated to date over existing KVSs including the best-published FPGA-based and GPU-based claims. We craft a set of design principles for future platform architectures, and via detailed simulations demonstrate the capability of achieving a billion RPS with a single server constructed following our principles.

Be Fast, Cheap and in Control with SwitchKV

Xiaozhou Li, Raghav Sethi, Michael Kaminsky, David G. Andersen & Michael J. Freedman.

In 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI'16), Santa Clara, CA, March 2016.

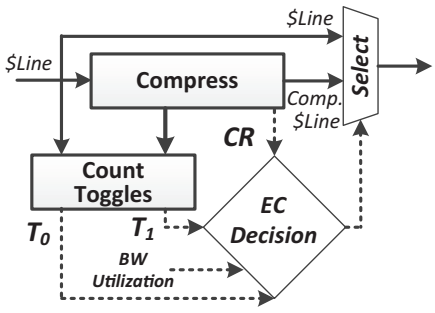
SwitchKV is a new key-value store system design that combines high-performance cache nodes with resource-constrained backend nodes to provide load balancing in the face of unpredictable workload skew. The cache nodes absorb the hottest queries so that no individual backend node is over-burdened. Compared with previous designs, SwitchKV exploits SDN techniques and deeply optimized switch hardware to enable efficient content-based routing. Programmable network switches keep track of cached keys and route requests to the appropriate nodes at line speed, based on keys encoded in packet headers. A new hybrid caching strategy keeps cache and switch forwarding rules updated with low overhead and ensures that system load is always well-balanced under rapidly changing workloads. Our evaluation results demonstrate that SwitchKV can achieve up to 5× throughput and 3× latency improvements over traditional system designs.

A Case for Toggle-Aware Compression for GPU Systems

Gennady Pekhimenko, Evgeny Bolotin, Nandita Vijaykumar, Onur Mutlu, Todd C. Mowry & Stephen W. Keckler

Proceedings of the 22nd International Symposium on High-Performance Computer Architecture (HPCA), Barcelona, Spain, March 2016.

Data compression can be an effective method to achieve higher system performance and energy efficiency in modern data-intensive applications by exploiting redundancy and data similarity. Prior works have studied a variety of data compression techniques to improve both capacity (e.g., of caches and main memory) and bandwidth utilization (e.g., of the on-chip and off-chip interconnects). In this paper, we make a new observation about the energy-efficiency of communication when compression is applied. While compression reduces the amount of transferred data, it leads to a substantial increase in the number of bit toggles (i.e., communication channel switchings from 0 to 1 or from 1 to 0). The increased toggle count increases the dynamic energy consumed by on-chip and off-chip buses due to more frequent charging and discharging of the wires. Our results show that the total bit toggle count can increase from 20% to 2.2× when compression is applied for some compression algorithms, averaged across different application suites. We characterize and demonstrate this new problem across 242 GPU applications and six different compression algorithms. To mitigate the problem, we propose two new toggle-aware compression techniques: Energy Control and Metadata Consolidation. These techniques greatly reduce the bit toggle count impact of the data compression algorithms we examine, while keeping most of their bandwidth reduction benefits.



Energy Control decision mechanism.

Towards Accurate and Fast Evaluation of Multi-Stage Log-Structured Designs.

Hyeontaek Lim, David G. Andersen & Michael Kaminsky

In 14th USENIX Conference on File and Storage Technologies (FAST'16), Santa Clara, CA, February 2016.

Multi-stage log-structured (MSLS) designs, such as LevelDB, RocksDB, HBase, and Cassandra, are a family of storage system designs that exploit the high sequential write speeds of hard disks and flash drives by using multiple append-only data structures. As a first step towards accurate and fast evaluation of MSLS, we propose new analytic primitives and MSLS design models that quickly give accurate performance estimates. Our model can almost perfectly estimate the cost of inserts in LevelDB, whereas the conventional worst-case analysis gives 1.8–3.5x higher estimates than the actual cost. A few minutes of offline analysis using our model can find optimized system parameters that decrease LevelDB's insert cost by up to 9.4–26.2%; our analytic primitives and model also suggest changes to RocksDB that re-

continued on page 18

April 2016

Joy Arulraj Receives Samsung PhD Fellowship

CMU DB Ph.D. student Joy Arulraj has won a Samsung 2017 PhD Fellowship in the area of Software and Memory System Solutions for Data Centers. Joy's research is on developing the novel database management system architectures for emerging non-volatile memory technologies to support modern hybrid transactional/analytical processing (HTAP) applications.

The Samsung PhD Fellowship program awards outstanding graduate students working on cutting-edge research for innovative solutions to their fields' biggest problems.

-- CMU Database Group News

April 2016

Alexey Tumanov and Team Win Best Student Paper at Eurosys16!

Congratulations to Alexey Tumanov, Timothy Zhu, Jun Woo Park, Michael A. Kozuch, and Mor Harchol-Balter, Gregory R. Ganger who have been awarded Best Student Paper for their work on "TetriSched: Global Rescheduling with Adaptive Plan-ahead in Dynamic Heterogeneous Clusters" at Eurosys16.

The paper describes TetriSched, a scheduler that works in tandem with a calendaring reservation system to continuously re-evaluate the immediate-term scheduling plan for all pending jobs on each scheduling cycle.

April 2016

Rajat Kateja Wins Best Graduate Poster Award!

The Award for the Carnegie Mellon Best Graduate Poster was presented to Rajat Kateja for his work on "Reducing Tail Latencies for Reads in Flash SSDs" on April 8th at the Carnegie Mellon Industry-Academia Partnership Workshop held at CMU. Participating professors and students were predominantly from the CMU School of Computer Science and the Department of Electrical and Computer Engineering, comprising members from many of the research centers, groups and labs including CyLab, Data Storage Systems Center, IoT Expedition, and the Parallel Data Lab. Rajat received a $300 cash award and framed certificate signed by his advisor and the IAP.

April 2016

Dana Van Aken Wins 2016 National Science Foundation Graduate Fellowship

CMU DB and PDL Ph.D. student Dana Van Aken won a National Science Foundation Graduate Fellowship. Dana's research is focused on using machine learning techniques for automatic database management system tuning and configuration.

NSF's Graduate Research Fellowship Program supports outstanding student researchers pursuing graduate degrees in science, technology, engineering and mathematics who demonstrate the potential to have a significant impact in their fields. Almost 17,000 students applied for a total of 2,000 fellowships awarded nationwide.

-- CMU Database Group News

March 2016

Kristy Gardner Receives SCS Graduate Student Teaching Award

Congratulations to Kristy Gardner on receiving the Alan J. Perlis Graduate Student Teaching Award for 2016. The SCS student teaching awards were inaugurated in 2005 and are named for Alan J. Perlis, a founder of the Computer Science Department at Carnegie Mellon and CMU's first Department Head (1965). These awards are based on student nominations, recommendation letters, and reviews and honors the student (graduate or undergraduate) who has shown the highest degree of excellence and dedication as a teaching assistant.

December 2015

Federal Trade Commission Appoints Lorrie Cranor as Chief Technologist

Federal Trade Commission Chairwoman Edith Ramirez has appointed Lorrie Faith Cranor as the agency's Chief Technologist, succeeding Ashkan Soltani. Cranor will join the FTC staff in January and be primarily responsible for advising Chairwoman Ramirez and the Commission on de-

continued on page 9

continued from page 8

veloping technology and policy matters.

Cranor is currently a Professor of Computer Science and Engineering and Public Policy at Carnegie Mellon University, where she directs the CyLab Usable Privacy and Security Laboratory. She was previously a researcher at AT&T Labs Research and has also taught at the Stern School of Business at New York University.

"Technology is playing an ever more important role in consumers' lives, whether through mobile devices, personal fitness trackers, or the increasing array of Internet-connected devices we find in homes and elsewhere," said FTC Chairwoman Edith Ramirez. "We are delighted to welcome Lorrie to our team, where she will play a key role in helping guide the many areas of FTC work involving new technologies and platforms.

Cranor has authored over 150 research papers on online privacy and usable security, and has played a central role in establishing the usable privacy and security research community, including her founding of the Symposium on Usable Privacy and Security. She is also a co-director of Carnegie Mellon's Privacy Engineering masters' program.

Cranor holds a doctorate in Engineering and Policy, masters' degrees in Computer Science, and Technology and Human Affairs, and a bachelor's degree in Engineering and Public Policy, from Washington University in St. Louis, Missouri.

The Federal Trade Commission works for consumers to prevent fraudulent, deceptive, and unfair business practices and to provide information to help spot, stop, and avoid them.

-- Dec. 3, 2015 FTC Press Release

October 2015
Welcome Layla!

Karen is proud to present her second grandchild, Layla Anne! Layla was born to Karen's daughter Laura and her husband Pete Losi on October 29, 2015, weighing 4 lbs. 13 oz.

September 2015
New PDL Faculty!

We welcome Phil Gibbons, as he joins CMU as a Professor in the Computer Science and Electrical & Computer Engineering Departments. Most recently Phil was a P.I. at the Intel Science and Technology Center for Cloud Computing (2011-2015) at CMU. Previous to this he was a researcher with the Intel Research Pittsburgh Lablet (2001-2011), the Information Sciences Research Center at Lucent Bell Laboratories (1996-2001), and the Mathematical Sciences Research Center at AT&T Bell Laboratories (1990-1996). His research areas include big data, parallel computing, databases, cloud computing, sensor networks, distributed systems and computer architecture. Phil received his Ph.D. in Computer Science from the University of California at Berkeley in 1989.

September 2015
Jiaqi Tan Wins FMCAD Award!

Congratulations Jiaqi Tan, for winning the Best Contribution Award at the Student Forum for the Formal Methods in Computer-Aided Design (FMCAD), held at the University of Texas, in Austin, TX, for his Ph.D. thesis work on "White-box Software Isolation with Fully Automated Black-box Proofs."

September 2015
Best Paper Award at MobiArch!

Congratulations to Utsav Drolia, Nathan Mickulicz, Rajeev Gandhi, and Priya Narasimhan on receiving the Best-Paper Award at the 10th ACM Workshop on Mobility in the Evolving Internet Architecture (MobiArch), held in Paris, France in September 2015. Their paper "Krowd: A Key-Value Store for Crowded Venues" proposes a novel way of developing a mobile infrastructure "by the people, for the people," through mobile-cloud clusters formed from mobile devices inside high-density environments such as sports stadiums.

September 2015
Samira Khan now Faculty at University of Virginia

Best wishes to Samira as she joins the CS department at the University of Virginia as an Assistant Professor. Samira is mainly interested in Computer Architecture and Computer systems, especially in building new systems by rethinking the traditional assumptions in abstraction and separation

continued on page 10

continued from page 9

of responsibilities in different system layers and redesigning interfaces with new interaction and collaboration to solve systems/architecture research problems.

September 2015

Three PDL Faculty Receive Google Faculty Research Awards

The Google Faculty Research Awards program aims to identify and support world-class, full-time faculty pursuing research in areas of mutual interest and are awarded twice a year. Congratulations to our three PDL faculty members who received the award for the Summer 2015 award term. Andy's work will focus on distributed, in-memory database management systems, Mor Harchol-Balter will be researching "When Many Workloads Share Networked Storage: How to Guarantee Tail Latency SLOs" (Google), and Lorrie Cranor will focus her award on research in the Human-Computer Interaction area.

September 2015

Two PDL Faculty Receive Facebook Faculty Awards

Congratulations to Andy Pavlo and Mor Harchol-Balter who each received a Facebook Faculty Award. Andy's research sponsored by the award will focus on distributed, in-memory database management systems. Mor will be investigating the "Performance Analysis and Design of Computer Systems."

August 2015

Best Paper Award at SoCC!

Congratulations to Jinliang Wei and co-authors Wei Dai, Aurick Qiao, Qirong Ho, Henggang Cui, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson, and Eric P. Xing on winning one of two awards for Best Paper at the 2015 ACM Symposium on Cloud Computing, held on the Kohala Coast, HI. Their paper Managed Communication and Consistency for Fast Data-Parallel Iterative Analytics presents Bösen, a system that maximizes network communication efficiency under a given inter-machine network bandwidth budget to minimize accumulated error, while ensuring theoretical convergence guarantees for large-scale data-parallel ML applications.

May 2015

Alexey Tumanov Receives ECE's Graduate Student Teaching Assistant Award

Congratulations to Alexey for receiving ECE's Outstanding Graduate Student Teaching Assistant Award for his efforts on 15-719: Advanced Cloud Computing, taught by Garth Gibson and Majd Sakr during the fall semester of 2014. In their letter of nomination, Professors Sakr and Gibson cited Alexey's hard work, innovation, and commitment to student success, describing it as "unparalleled". Alexey "went way beyond the call of duty, supported the students with a pleasant constructive engagement style and built a project [that] will certainly [be] reused next year."

During the semester, Alexey developed the end-of-term course project, where the students were guided to build their own virtualized clusters and cluster schedulers on the brand new PRObE cluster called NOME. In the words of one of the students: "[Alexey was] extremely helpful and responsive. [We] had a lot of one-on-one discussions, which led to interesting insights and learning. [He] was very supportive of ideas and any issues faced. [He] strived hard to get the essence of the project into the students and drive the phases towards that goal. Probably my best project at CMU."

-- with info from D. Marculescu's award presentation notes.

May 2015

NVIDIA Graduate Fellowship Winner

Congratulations to Gennady Pekhimenko on receiving an NVIDIA Graduate Fellowship. Recipients are selected based on their academic achievements, professor nomination, and area of research. Gennady's general research focus is on energy-efficient memory systems using hardware-based data compression. He discovered a series of mechanisms that exploit the existing redundancy in applications' data to perform efficient compression in caches and main memory, thereby providing higher effective capacity and higher available bandwidth across the memory hierarchy. His most recent work is looking into how to perform energy-efficient bandwidth compression for modern GPUs. Gennady is advised by he is advised by Todd Mowry and Onur Mutlu.

SIMILARITY-BASED DEDUPLICATION FOR DATABASES

Lianghong Xu, Andrew Pavlo, Sudipta Sengupta & Gregory R. Ganger

The rate of data growth continues to outpace the decline of hardware costs. One solution to this problem is database compression. In addition to saving space, database compression methods help reduce the number of disk I/Os and improve performance, since queried data fits into fewer pages. For distributed databases replicated across geographical regions, there is also a strong need to reduce the amount of data transfer used to keep replicas in sync.

The most widely used approach for data reduction in database management systems (DBMSs) is block-level compression. Although this method is simple and effective, it fails to address redundancy across blocks and therefore leaves significant room for improvement in many applications (e.g., due to app-level versioning in wikis or partial record copying in message boards). Deduplication (dedup) has become popular in backup systems for eliminating duplicate content across an entire data corpus, and often achieves much higher compression ratios. The backup stream is divided into chunks, and a collision-resistant hash (e.g., SHA-1) is used as each chunk's identity. The dedup system maintains a global index of all hashes and uses it to detect duplicates. Dedup works well for both primary and backup storage data sets comprised of large files that are rarely modified (and if they are modified, the changes are sparse).

Unfortunately, traditional chunk-based dedup schemes are unsuitable for operational DBMSs, where many update queries modify a single record. The duplicate data in records is too fine-grained unless the system uses very small chunk sizes. But, relatively large chunk sizes (e.g., 4–8 KB) are the norm to avoid huge in-memory indices and large numbers of disk reads.

Our research develops dbDedup, a lightweight scheme for on-line database systems that uses similarity-based deduplication [1] to compress individual records. This reduces the size of data stored on disk beyond what traditional compression schemes provide. Instead of indexing every chunk hash, dbDedup samples a small subset of chunk hashes for each new database record and uses this sample to identify a similar record in the database. It then uses byte-level delta compression on the two records to reduce both online storage used and remote replication bandwidth by providing higher compression ratios with lower memory overhead than chunk-based dedup. It also combines well with block-level compression, as illustrated in Figure 1.

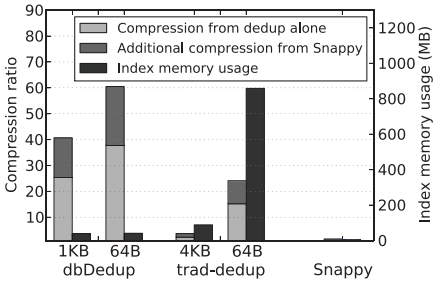


Figure 1: Compression ratio and index memory usage for Wikipedia data stored in five MongoDB configurations: with dbDedup (1KB chunk size and 64B), with traditional dedup (4KB and 64B), and with Snappy (block-level compression). dbDedup provides higher compression ratio and lower index memory overhead than traditional dedup. Snappy provides the same 1:6X compression for the post-dedup data or the original data.

We implemented dbDedup in the MongoDB DBMS (www.mongodb.org) and evaluated its efficacy using four real-world datasets. Our results show that it achieves up to 37X reduction (61X when combined with block-level compression) in storage size and replication traffic, significantly outperforming chunk-based dedup, while imposing negligible impact on the DBMS's runtime performance.

In addition to borrowing sampling-based and cache-aware approaches to selecting source records from our recent sDedup system [1], dbDedup introduces and combines several novel techniques in order to achieve such efficiency. It uses novel two-way encoding to efficiently transfer encoded new records (forward encoding) to remote replicas, while storing unencoded new records with encoded forms of selected source records (backward encoding). As a result, no decode is required in the common case of accessing the most recent record in an encoding chain (e.g., the latest Wikipedia version). To avoid performance overhead from updating source records, dbDedup introduces a lossy write-back delta cache tuned to maximize the compression ratio while avoiding I/O contention. dbDedup also uses a new technique called hop encoding to minimize the worst-case number of decode steps required to access a specific record in a long encoding chain. Finally, dbDedup adaptively skips dedup efforts for databases and records where little savings are expected. Overall, dbDedup provides benefits with negligible effect on DBMS throughput or client latency (average and tail). Please see the PDL technical report [2] for more information.

References

[1] Reducing Replication Bandwidth for Distributed Document Databases. Lianghong Xu, Andy Pavlo, Sudipta Sengupta, Jin Li, and Gregory R. Ganger. In SoCC, pages 222–235, 2015.

[2] Similarity-based Deduplication for Databases. Lianghong Xu, Andrew Pavlo, Sudipta Sengupta, Gregory R. Ganger. Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-16-101, April 2016.

BIG-LEARNING SYSTEMS FOR BIG DATA

continued from page 1

among threads, and a next amount of work is executed. Commonly, in iterative ML, one iteration over the training data is performed between each pair of barriers.

Early parallel ML implementations used direct message passing among threads for coordination and update exchanges, forcing the ML application writer to deal with all of the complexities of parallel computing. The rise of map-reduce-style data processing systems, like Hadoop and Spark, allowed simpler implementations but suffer significant performance problems for parallel ML due to strict limitations on state sharing and communication among threads.

The most efficient modern frameworks for parallel ML use a *parameter server* architecture to make it easier for ML programmers to build and scale ML applications ("More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server" [1], "Scaling Distributed Machine Learning with the Parameter Server" [2]). Figure 1 illustrates this architecture in which all state shared among worker threads is kept in a specialized key-value store, which is commonly sharded across the same machines used to execute the worker threads. Worker threads process their assigned training data using simple READ and UPDATE methods to check and adjust parameter values. UPDATE operations must be sufficiently commutative and associative that concurrent UPDATEs by different workers can be applied to the shared parameters in any order. To avoid constant remote communication, the client-side library caches parameter values locally and buffers updates.

Parameter Server Specialization

PDL has a strong track record of innovation on efficient, high-performance key-value store systems, such as FAWN, SILT, and MICA. Because a parameter server is a key-value store, much of PDL's previous work applies. But, we have also found that unique characteristics of parallel ML invite specialization, such as to exploit the iterative-ness and convergence properties to increase efficiency and complete training more quickly.

Ben Blum talks about his research on "Concurrency Testing with Data-Race Preemption Points" at the 2015 PDL Retreat.

Bounded Staleness. The convergence property allows a good solution to be found from any initial guess, as described above. This same property ensures that minor errors in the adjustments made by any given iteration will not prevent success. This imperfection tolerance can be exploited to improve performance by allowing parallel and distributed threads to use looser consistency models for shared state (i.e., the current solution). But, there must be limits to ensure success. We have developed new consistency models in which each thread works with a view of the current solution that may not reflect all updates from other threads. Allowing such *staleness* reduces communication costs (batched updates and cached reads) and synchronization (less waiting for locks or straggling threads). Our new flexible model, called Stale Synchronous Parallel (SSP), avoids barriers and allows threads to be a bounded number (the *slack*) of iterations ahead of the current slowest thread. The maximum amount of staleness allowed, dictated by a tunable slack parameter, controls a tradeoff between time-per-iteration and quality-per-iteration… the optimum value balances the two to minimize overall time to convergence. Both proofs and experiments demonstrate the effectiveness of SSP in safely improving convergence speed. For more information, see "Scaling Distributed Machine Learning with the Parameter Server" [1] and "Exploiting Bounded Staleness to Speed up Big Data Analytics" [3].

Exploiting Iterative-ness. The iterative-ness property creates an opportunity: knowable repeating patterns of access to the shared state (i.e., current parameter values). Often, each thread processes its portion of the training data in the same order in each iteration, and the same subset of parameters are read and updated any time a particular training data item is processed. So, each iteration involves the same pattern of reads and writes to the shared state. Our IterStore system demonstrates how the repeated patterns can be discovered efficiently and exploited to greatly improve efficiency, both within a multi-core machine and for communication across machines. Examples include replacing dynamic cache and server structures with static pre-serialized structures, informing prefetch and partitioning decisions, and determining which data should be cached at each thread to avoid both contention and slow accesses to memory banks attached to other sockets. We found that such specializations reduce per-iteration runtimes by 33-98%. For more information, see "Exploiting Iterative-ness for Parallel ML Computations" [4].

continued on page 13

continued from page 13

Managed Communication. Bounded staleness ensures a worst-case limit on the staleness observed by any worker, allowing convergence to be assured, but does not address the average staleness. Because staleness can result in imperfect parameter updates, which reduces quality-per-iteration and thereby requires more iterations, there is value in reducing it. For example, proactive propagation of updates reduces average staleness, but can reduce performance when network bandwidth is limited. Our Bösen system demonstrates how to maximize effective usage of a given network bandwidth budget. Through explicit bandwidth management, Bösen fully utilizes, but never exceeds, the identified bandwidth availability to communicate updates as aggressively as possible. Moreover, Bösen prioritizes use of limited bandwidth for messages that most affect algorithm convergence (e.g., those with the most significant updates). We found that such bandwidth management greatly improves convergence speeds and makes algorithm performance more robust to different cluster configurations and circumstances. For more information, see "Managed Communication and Consistency for Fast Data-Parallel Iterative Analytics" [5].

GPU Specialization. Large-scale deep learning, which is becoming increasingly popular for ML on visual and audio media, requires huge computational resources to train multi-layer neural networks. The computation involved can be done much more efficiently on GPUs than on traditional CPU cores, but training on a single GPU is too slow and training on distributed GPUs can be inefficient due to data movement overheads, GPU stalls, and limited GPU memory. Our GeePS system demonstrates specializations that allow a parameter server system to efficiently support parallel ML on distributed GPUs, overcoming these obstacles. For example, it uses pre-built indexes to enable parallel fetches and updates, GeePS-managed caching in GPU memory, and GPU-friendly background data staging. It also exploits iterative-ness observations and the layered nature of neural networks to stage data between GPU and CPU memories to allow training of very large models, overcoming GPU memory size limits. Experiments show excellent scaling across distributed GPUs. For more information, see "GeePS: Scalable Deep Learning on Distributed GPUs with a GPU-Specialized Parameter Server" [6].

Brad Settlemyer (L), Gary Grider (CL), Carolyn Connor (R), all with Los Alamos National Laboratory, visit with Garth Gibson (CR) at the 2015 PDL Retreat.

Work Distribution and Scheduling

Most parallel ML implementations use a *data-parallel* approach. The training data is divided among worker threads that execute in parallel, each performing the work associated with their shard of the training data and communicating updates after completing each iteration over that shard. Typically, the assignment of work to workers stays the same from one iteration to the next. This continuity enables significant efficiency benefits, relative to independently scheduling work each iteration, in addition to avoiding potential scalability and latency challenges of a central scheduler. The efficiency benefits come from cache affinity effects (e.g., of the large training data) as well as from exploiting iterative-ness, yielding large benefits. But, we have developed ML-specific approaches to better distribute and schedule work among workers so as to adapt to dynamic cluster and ML algorithm behavior.

Straggler Mitigation. Parallel ML can suffer significant performance losses to stragglers. A straggler problem occurs when worker threads experience uncorrelated performance jitter. Each time synchronization is required, any one slowed worker thread can cause significant unproductive wait time for the others. Unfortunately, each if the load is balanced, transient slowdowns are common in real systems and have many causes, such as resource contention, garbage collection, background OS activities, and (for ML) stopping criteria calculations. Worse, the frequency of such issues rises significantly when executing on shared computing infrastructures rather than dedicated clusters and as the number of machines increases. Our FlexRR system demonstrates a solution to the straggler problem for iterative convergent ML. With temporary work reassignment, a slowed worker can offload a portion of its work for an iteration to workers that are currently faster, helping the slowed worker catch up. FlexRR's specialized reassignment scheme complements bounded staleness, and both are needed to solve the straggler problem. Flexible consistency via bounded staleness provides FlexRR with the extra time needed to detect slowed workers and address them with temporary work reassignment, before any worker reaches the bound and is blocked. Experiments on Amazon EC2 and local clusters confirm FlexRR's effectiveness. For more information,

continued on page 27

DISSERTATION ABSTRACT:
Similarity-based Deduplication for Databases

Lianghong Xu

Carnegie Mellon University, ECE

Ph.D. Defense
April 21, 2016

The rate of data growth outpaces the decline of hardware costs, and there has been an ever-increasing demand in reducing the storage and network overhead for online database management systems (DBMSs). The most widely used approach for data reduction in DBMSs is block- level compression. Although this method is simple and effective, it fails to address redundancy across blocks and therefore leaves significant room for improvement for many applications.

This dissertation proposes a systematic approach, termed similarity-based deduplication, which reduces the amount of data stored on disk and transmitted over the network beyond the benefits provided by traditional compression schemes. To demonstrate the approach, we designed and implemented dbDedup, a lightweight record-level similarity-based deduplication engine for online DBMSs. The design of dbDedup exploits key observations we find in database workloads, including small item sizes, temporal locality, and the incremental nature of record updates. The proposed approach fundamentally differs from traditional chunk-based deduplication approaches in that, instead of finding identical chunks anywhere else in the data corpus, similarity-based deduplication identifies a single similar data-item and performs differential compression to remove the redundant parts for greater savings.

To achieve high efficiency, dbDedup introduces novel encoding, caching and similarity selection techniques that significantly mitigate the deduplication overhead with minimal loss of compression ratio. To evaluate our work, we integrated dbDedup into the storage and replication components of a distributed NoSQL DBMS and analyzed its properties using four real datasets. Our results show that dbDedup achieves up to 37× reduction in the storage size and replication traffic of the database on its own and up to 61× reduction when paired with the DBMS's block-level compression. dbDedup provides both benefits with negligible effect on DBMS throughput or client latency (average and tail).

Lin Xiao talks about her research on "ShardFS vs. IndexFS: Replication vs. Caching Strategies in Cloud Storage Systems" with Roland Wunderlich (Two Sigma) and Kim Keeton (HP Labs).

DISSERTATION ABSTRACT:
Simple DRAM and Virtual Memory Abstractions to Enable Highly Efficient Memory Subsystems

Vivek Seshadri

Carnegie Mellon University, SCS

Ph.D. Defense
March 21, 2016

In most modern systems, the memory subsystem is managed and accessed at multiple different granularities (e.g., words, cache lines, and pages) at various resources. In this thesis, we observe that such multi-granularity management results in significant inefficiency in the memory subsystem. Specifically, we observe that 1) page-granularity virtual memory unnecessarily triggers large memory operations, and 2) existing cache-line granularity off-chip memory interface is inefficient for performing bulk data operations and operations that exhibit poor spatial locality. To address these problems, we present a series of techniques in this thesis.

First, to address the inefficiency of existing page-granularity virtual memory systems, we propose a new framework called page overlays. At a high level, our framework augments the existing virtual memory framework with the ability to track a new version of a subset of cache lines within each virtual page. We show that this simple extension is powerful by demonstrating its benefits on a number of different applications.

DISSERTATION ABSTRACT:
Black-Box Problem Diagnosis in Parallel File Systems

Michael P. Kasick

Carnegie Mellon University, SCS

Ph.D. Defense
December 10, 2015

Parallel file systems target large, high-performance storage systems. Since these storage systems are comprised of a significant number of components (i.e., hundreds of file servers, thousands of disks, etc.), they are expected to (and in practice do) frequently exhibit "problems", from degraded performance to outright failure of one or more components. The sheer number of components, and thus, potential problems, makes manual diagnosis of these problems difficult. Of particular concern are system-wide performance degradations, which may arise from a single misbehaving component, and thus, pose a challenge for problem localization. Even failure of a redundant component with a less-significant performance impact is worrisome as it may, in absence of explicit checks, go

continued on page 15

continued from page 14

unnoticed for some time and increase risk of system unavailability.

As a solution, this thesis defines a novel problem-diagnosis approach, capitalizing upon the parallel-file-system design criterion of balanced performance, that peer-compares the performance of system components to diagnoses problems within storage systems running unmodified, "off-the-shelf" parallel file systems. Performed in support of this thesis is a set of laboratory experiments that demonstrate proof-of-concept of the peer-comparison approach by injecting four realistic problems into 12-server, test-bench PVFS and Lustre clusters. This thesis is further validated by taking the diagnosis approach and adapting it to work on a very-large, production GPFS storage system consisting of 128 file servers, 32 storage controllers, 1152 disk arrays, and 11,520 total disks. Presented in a 15-month case study is the problems observed through analysis of 624 GB of instrumentation data, in which a variety of performance-related storage-system problems are diagnosed, in a matter of hours, as compared to the days or longer with manual approaches.

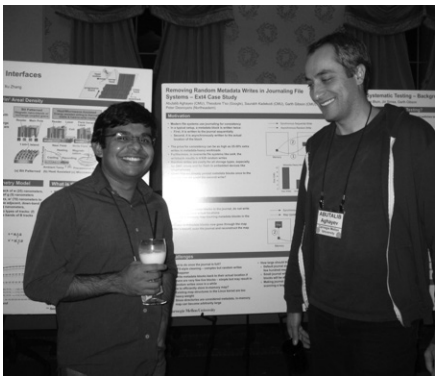
DISSERTATION ABSTRACT:
Liberating Virtual Machines from Physical Boundaries through Execution Knowledge

Yoshihisa Abe

Carnegie Mellon University, SCS

Ph.D. Defense
December 9, 2015

Hardware virtualization enables remote instantiation of computation through the preserved executability of encapsulated software. The large size of virtual machines (VMs), however, poses challenges in exploiting this strong feature under the existence of resource constraints. In this thesis, we claim that the use of execution knowledge achieves the efficiency and timeliness of VM state transfer in such environments. We demonstrate its effectiveness in two concrete contexts in which the challenges materialize: 1) VM delivery over WANs, with network resource limitations, and 2) urgent migration of VMs under contention, with strict time requirements. In the context of VM delivery over WANs, we take advantage of the knowledge about past VM execution instances. We conduct the evaluation of vTube, a system for efficiently streaming virtual appliances, from both systems and human-centric perspectives. In the context of urgent migration of VMs under contention, we leverage current execution knowledge at the guest OS level. Our approach, called enlightened post-copy, uses this knowledge to expedite the resolution of contention between VMs. Our proposed solutions address the corresponding problems by providing VM performance as defined by critical metrics in their specific contexts.



Saurabh Kadekodi and Abutalib Aghayev are ready to discuss their work on Shingled Disks and Writes in Journaling File Systems at the 2015 PDL Retreat.

DISSERTATION ABSTRACT:
Agentless Cloud-wide Monitoring of Virtual Disk State

Wolfgang Richter

Carnegie Mellon University, SCS

Ph.D. Defense
September 18, 2015

This dissertation proposes a fundamentally different way of monitoring persistent storage. It introduces a monitoring platform based on the modern reality of software defined storage which enables the decoupling of policy from mechanism. The proposed platform is both agentless—meaning it operates external to and independent of the entities it monitors—and scalable—meaning it is designed to address many systems at once with a mixture of operating systems and applications. Concretely, this dissertation focuses on virtualized clouds, but the proposed monitoring platform generalizes to any form of persistent storage.

The core mechanism this dissertation introduces is called Distributed Streaming Virtual Machine Introspection (DS-VMI), and it leverages two properties of modern clouds: virtualized servers managed by hypervisors enabling efficient introspection, and file-level duplication of data within cloud instances. We explore a new class of agentless monitoring applications via three interfaces with two different consistency models: \cloudinotify (strong consistency), \slashcloud (eventual consistency), and\slashhistory (strong consistency). \cloudinotify is a publish-subscribe interface to cloud-wide file-level updates and it supports event-based monitoring applications.\slashcloud is designed to support batch-based and legacy monitoring applications by providing a file system interface to cloud-wide file-level state. \slashhistory is designed to support efficient search and management of historic virtual disk state. It leverages new fast-to-access archival storage systems, and achieves tractable indexing of file-level history via whole-file deduplication using a novel application of an incremental hashing construction.

continued on page 16

continued from page 15

DISSERTATION ABSTRACT:
Resource-Efficient Data-Intensive System Designs for High Performance and Capacity

Hyeontaek Lim

Carnegie Mellon University, SCS

Ph.D. Defense
July 20, 2015

Data-intensive systems are a critical building block of today's large-scale Internet services. These systems have enabled high throughput and capacity, reaching billions of requests per second for trillions of items in a single storage cluster. However, many systems exhibit a large amount of inefficiencies; for instance, memcached, a widely-used in-memory key-value store system, handles 1--2 million requests per second on a modern server node, whereas an optimized software system could achieve over 70 million requests per second using the same hardware. Reducing such inefficiencies can improve the cost effectiveness of the systems significantly.

This dissertation shows that by leveraging modern hardware and exploiting workload characteristics, data-intensive storage systems that process a large amount of fine-grained data can achieve an order of magnitude higher performance and capacity than prior systems that are built for generic hardware and workloads. As examples, we present SILT and MICA, which are resource-efficient key-value stores for flash and memory. SILT provides flash-speed query processing and 5.7X higher capacity than the previous state-of-the-art system. It employs new memory-efficient indexing schemes including ECT that requires only 2.5 bits per item in memory, and a system cost model built upon new accurate and fast analytic primitives to find workload-specific system configurations. MICA offers 4X higher throughput over the network than previous in-memory key-value store systems by performing efficient parallel request processing on multi-core processors and low-overhead request direction with modern network interface cards, and by using new key-value data structures designed for specific workload types.

THESIS PROPOSAL:
Analyzing Systems with Redundant Requests

Kristen Gardner, SCS

December 14, 2015

Reducing response time is a primary focus of computer systems research. One key factor that influences a job's response time in a multi-server system is dispatching: when a job arrives to the system, it must be sent immediately to one of the servers. This thesis focuses on a new dispatching policy: redundancy. Unlike traditional dispatching policies, which send only a single copy of each job, the idea of redundancy is to dispatch multiple copies of the same job and wait for the first copy to complete service. A great deal of empirical work has demonstrated that redundancy can provide substantial response time improvements. For example, using redundancy in MapReduce systems has been shown to reduce the response time of straggling tasks by 20 – 50%. However, despite the extensive empirical studies on redundancy, there has been very little theoretical work analyzing performance in redundancy systems.

In this thesis, we propose to (1) quantify the benefits and costs of redundancy by providing the first analysis of systems with redundancy, and (2) design better redundancy systems to reduce response time. This proposal begins by reviewing our completed work, in which we develop exact analysis of response time in an idealized redundancy model. We use this analysis to explore initial messages on the benefits and costs of redundancy in small systems; we then analyze response time in systems with large numbers of servers. Our proposed future work consists of three major foci: developing and analyzing theoretical models of redundancy that capture the characteristics of real systems; investigating how smart scheduling policies can be used to improve performance in redundancy systems; and exploring applications for redundancy systems.

THESIS PROPOSAL:
Provable, Programmer-Visible Control-Flow Integrity for Software

Jiaqi Tan, SCS

December 1, 2015

Control-Flow Integrity (CFI) is an important safety and security property of software that is especially important in safety-critical systems, such as medical devices and flight-control systems, where failures can lead to catastrophic accidents and even the loss of lives. While current techniques can both provide and verify CFI, they either (i) do so entirely at the machine-code level, where CFI mechanisms cannot be observed by programmers, and may introduce unexpected changes to the software, which is highly undesirable for safety-critical software, or (ii) they do so entirely at the source-code level, where CFI verification is not automated, and requires programmers to manually provide specialized verification inputs such as loop invariants, safety assertions, and function contracts. In this dissertation, I propose to develop a novel approach to provide CFI for software that both (i) can be formally proven in a fully- automatic way, and (ii) provides CFI mechanisms that are programmer-visible. The proposed approach in my dissertation to CFI consists of three steps: (i) formally proving CFI at the machine-code level using a novel logic framework that I

continued on page 17

continued from page 16

develop which automates CFI proofs for ARM machine-code programs, (ii) automatically identifying culprit machine-code instructions responsible for CFI proof failures, and (iii) automatically generating source-code hints for programmers to remedy CFI proof failures in their programs. This enables CFI to be achieved in a way that strong, tangible evidence in the form of formal CFI proofs can be obtained without specialized inputs from programmers, and in a way that programmers can consider the meaning and behavior of their programs in implementing CFI mechanisms for their programs, thus facilitating CFI for software in a way that is amenable to safety-critical systems.

THESIS PROPOSAL:
Meeting Tail Latency SLOs in Shared Networked Storage

Timothy Zhu, SCS

November 17, 2015

Meeting tail latency Service Level Objectives (SLOs) in shared networked storage systems is an important and challenging problem in datacenters. Our work is motivated by three trends:

First, companies like Google and Amazon are increasingly interested in long tails at the 99th and 99.9th percentile latencies. As technology improves, users are more accustomed to low latency and start to expect near instant response times. Furthermore, as workloads become more parallel, the need for low tail latencies becomes increasingly important since jobs often run at the speed of the slowest request.

Aaron Harlap presents his research on "Addressing the Straggler Problem in Iterative Convergent Parallel ML" at the 2015 PDL Retreat.

Second, as workloads become increasingly data-driven, I/O latencies due to storage and networks play a large part in the end-to-end user experience for latency sensitive applications. Storage is often the hardest resource to share and is typically the bottleneck resource. Unless storage can be completely avoided, storage latencies typically have the most impact on overall latency, particularly at the tail.

Third, workloads are moving into multi-tenant cloud environments where resources are shared, particularly network and storage. This shift in industry to consolidate workloads onto shared public and private clouds is beneficial in reducing resource and management costs of computing infrastructures. However, while consolidation leads to greater economies of scale, it also introduces challenges in meeting tail latency SLOs.

In our proposed work, we will demonstrate that we can build a networked storage system that can meet tail latency SLOs for many workloads sharing the system. Specifically, we will study how to meet tail latency SLOs from the perspective of scheduling policies, admission control, and workload placement/migration.

THESIS PROPOSAL:
Better End-to-End Adaptation Using Centralized Predictive Control

Junchen Jiang, SCS

July 2, 2015

Transport layer and application layer of network stack use end-to-end adaptation protocols (e.g., TCP and bitrate-adaptive video) to achieve high performance by continuously adapting endpoint behavior to changes of network conditions. The traditional belief is that these protocols must be run independently by endpoints to achieve desirable performance. In essence, they use reactive logic triggered only by locally observable events. For instance, TCP reacts to a packet timeout by halving the congestion window.

In this thesis, we argue that centralized predictive control can lead to better end-to-end adaptation and large performance improvement at both transport layer and application layer. We show that it is feasible to decouple adaptation logics from end-to-end adaptation protocols and centralize them into a global controller that makes predictive control using a global view of different connections' performance. For instance, TCP with centralized predictive control can predict the best congestion window using other similar TCP sessions' performance.

To deliver the promised performance benefits of centralized predictive control, we must address two key technical challenges. First, we present prediction algorithms, which accurately predict the optimal adaptation behavior of endpoints by exploiting the structural information of the global view (e.g., some connections are subjected to same network bottleneck). Second, we present designs of a scalable control platform, which leverage the persistence of optimal decisions to minimize negative impacts of the inherent delay between the controller and widely distributed endpoints.

This thesis will present algorithms and system designs of centralized predictive control for both transport layer and application layer. We show that our ap-

continued on page 18

DEFENSES & PROPOSALS

continued from page 17

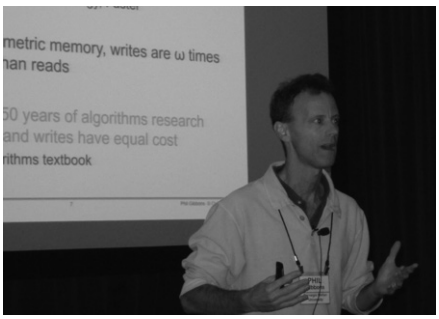
proach can lead to better performance for TCP, Internet video and real-time communication applications like Skype. Our preliminary experiments have shown significant improvement of Internet video quality by centralized predictive control.

THESIS PROPOSAL:
Scheduling with Space-Time Soft Constraints in Heterogeneous Cloud Datacenters

Alexey Tumanov, ECE

May 2015

Heterogeneity in datacenter hardware, software, and user objectives calls for new scheduling schemes to capture, aggregate, and leverage this information. Our proposed scheduler, TetriSched, explicitly considers cluster job-specific preferences in terms of where (space), when(time), and how (space-time shape) these jobs are scheduled. Spatial and temporal preferences combined allow TetriSched to provide higher overall value to complex data analytics mixes consolidated on heterogeneous collections of resources. First, we propose a principal building block—a new language called Space-Time Request Language (STRL). It enables the expression of these preferences in a general, extensible way by using a declarative, composable, algebraic structure with combinatorial primitives and allows TetriSched to understand which resources are preferred and by how much, over other acceptable options. Estimated job runtimes for recurrent or profiled jobs allow TetriSched to consider deferred placement if the benefit of waiting for unavailable preferred resources exceeds the cost. Second, building on the generality of STRL, we propose an equally general STRL Compiler that automatically compiles STRL expressions into Mixed Integer Linear Programming (MILP) problems that can be aggregated and solved to maximize the overall value of shared cluster resources. Third, we propose a set of features that extend the scope and the practicality of TetriSched's deployment by analyzing and improving on its scalability, enabling and studying the efficacy of preemption, and featuring sub-machine granularity resource assignment within a single scheduling cycle. The first set of experiments with a variety of job type mixes, workload intensities, degrees of burstiness, preference strengths, and input inaccuracies support our hypothesis that leveraging space-time soft constraints is (a) beneficial and (b) possible to achieve.



Phil Gibbons, new PDL Faculty member, talks about "Write-efficient Algorithms for Emerging Memory Technologies" at the 2015 PDL Retreat.

RECENT PUBLICATIONS

continued from page 7

duce its insert cost by up to 32.0%, without reducing query performance or requiring extra memory.

Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM

Kevin K. Chang, Prashant J. Nair, Donghyuk Lee, Saugata Ghose, Moinuddin K. Qureshi & Onur Mutlu

Proceedings of the 22nd International Symposium on High-Performance Computer Architecture (HPCA), Barcelona, Spain, March 2016.

This paper introduces a new DRAM design that enables fast and energy-efficient bulk data movement across subarrays in a DRAM chip. While bulk data movement is a key operation in many applications and operating systems, contemporary systems perform this movement inefficiently, by transferring data from DRAM to the processor, and then back to DRAM, across a narrow off-chip channel. The use of this narrow channel for bulk data movement results in high latency and energy consumption. Prior work proposed to avoid these high costs by exploiting the existing wide internal DRAM bandwidth for bulk data movement, but the limited connectivity of wires within DRAM allows fast data movement within only a single DRAM subarray. Each subarray is only a few megabytes in size, greatly restricting the range over which fast bulk data movement can happen within DRAM.

We propose a new DRAM substrate, Low-Cost Inter-Linked Subarrays (LISA), whose goal is to enable fast and efficient data movement across a large range of memory at low cost. LISA adds low-cost connections between adjacent subarrays. By using these connections to interconnect the existing internal wires (bitlines) of adjacent subarrays, LISA enables wide-bandwidth data transfer across multiple subarrays with little (only 0.8%) DRAM area

continued on page 19

continued from page 18

overhead. As a DRAM substrate, LISA is versatile, enabling an array of new applications. We describe and evaluate three such applications in detail: (1) fast inter-subarray bulk data copy, (2) in-DRAM caching using a DRAM architecture whose rows have heterogeneous access latencies, and (3) accelerated bitline precharging by linking multiple precharge units together. Our extensive evaluations show that each of LISA's three applications significantly improves performance and memory energy efficiency, and their combined benefit is higher than the benefit of each alone, on a variety of workloads and system configurations.

Simultaneous Multi-Layer Access: Improving 3D-Stacked Memory Bandwidth at Low Cost

Donghyuk Lee, Saugata Ghose, Gennady Pekhimenko, Samira Khan & Onur Mutlu

ACM Transactions on Architecture and Code Optimization (TACO), Vol. 12, January 2016. Presented at the 11th HiPEAC Conference, Prague, Czech Republic, January 2016.

3D-stacked DRAM alleviates the limited memory bandwidth bottleneck that exists in modern systems by leveraging through silicon vias (TSVs) to deliver higher external memory channel bandwidth. Today's systems, however, cannot fully utilize the higher bandwidth offered by TSVs, due to the limited internal bandwidth within each layer of the 3D-stacked DRAM. We identify that the bottleneck to enabling higher bandwidth in 3D-stacked DRAM is now the global bitline interface, the connection between the DRAM row buffer and the peripheral IO circuits. The global bitline interface consists of a limited and expensive set of wires and structures, called global bitlines and global sense amplifiers, whose high cost makes it difficult to simply scale up the bandwidth of the interface within a single DRAM layer in the 3D stack. We alleviate this bandwidth bottleneck by exploiting the observation that several global bitline interfaces already exist across the multiple DRAM layers in current 3D-stacked designs, but only a fraction of them are enabled at the same time.

We propose a new 3D-stacked DRAM architecture, called Simultaneous Multi-Layer Access (SMLA), which increases the internal DRAM bandwidth by accessing multiple DRAM layers concurrently, thus making much greater use of the bandwidth that the TSVs offer. To avoid channel contention, the DRAM layers must coordinate with each other when simultaneously transferring data. We propose two approaches to coordination, both of which deliver four times the bandwidth for a four-layer DRAM, over a baseline that accesses only one layer at a time. Our first approach, Dedicated-IO, statically partitions the TSVs by assigning each layer to a dedicated set of TSVs that operate at a higher frequency. Unfortunately, Dedicated-IO requires a non-uniform design for each layer (increasing manufacturing costs), and its DRAM energy consumption scales linearly with the number of layers. Our second approach, Cascaded-IO, solves both issues by instead time multiplexing all of the TSVs across layers. Cascaded-IO reduces DRAM energy consumption by lowering the operating frequency of higher layers. Our evaluations show that SMLA provides significant performance improvement and energy reduction across a variety of workloads (55%/18% on average for multiprogrammed workloads, respectively) over a baseline 3D-stacked DRAM, with low overhead.

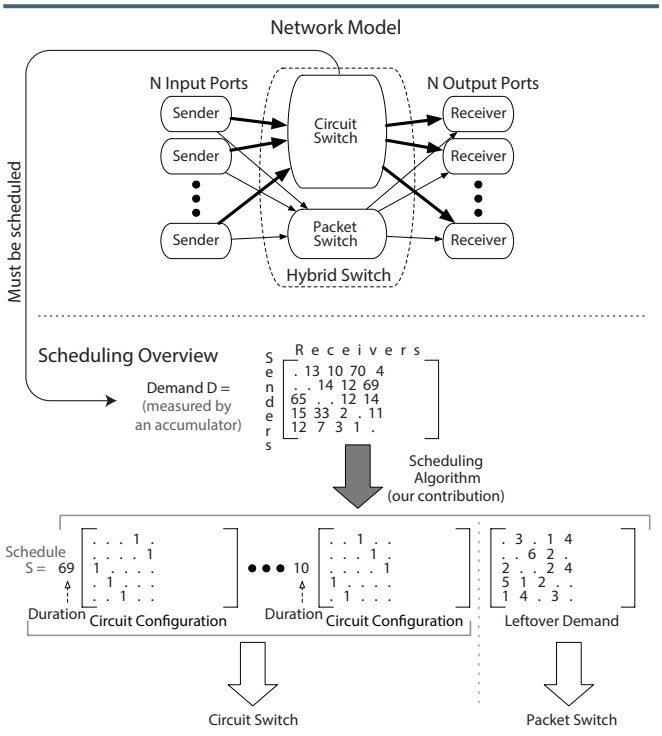
Scheduling Techniques for Hybrid Circuit/Packet Networks

He Liu, Matthew K. Mukerjee, Conglong Li, Nicolas Feltman, George Papen, Stefan Savage, Srinivasan Seshan, Geoffrey M. Voelker, David G. Andersen, Michael Kaminsky, George Porter & Alex C. Snoeren

The 11th International Conference on emerging Networking EXperiments and Technologies (CoNEXT 2015), Heidelberg, Germany, December 2015.

A range of new datacenter switch designs combine wireless or optical circuit technologies with electrical packet switching to deliver higher performance at lower cost than traditional packet-switched networks. These "hybrid" networks schedule large traffic demands

continued on page 20



Our model of a hybrid switch architecture and the scheduling process. The circuit switch has high bandwidth, but slow reconfiguration time. The packet switch has low bandwidth (e.g., an order of magnitude lower), but can make forwarding decisions per-packet.

continued from page 19

via a high-rate circuits and remaining traffic with a lower-rate, traditional packet-switches. Achieving high utilization requires an efficient scheduling algorithm that can compute proper circuit configurations and balance traffic across the switches. Recent proposals, however, provide no such algorithm and rely on an omniscient oracle to compute optimal switch configurations.

Finding the right balance of circuit and packet switch use is difficult: circuits must be reconfigured to serve different demands, incurring non-trivial switching delay, while the packet switch is bandwidth constrained. Adapting existing crossbar scheduling algorithms proves challenging with these constraints. In this paper, we formalize the hybrid switching problem, explore the design space of scheduling algorithms, and provide insight on using such algorithms in practice. We propose a heuristic-based algorithm, Solstice that provides a 2.9X increase in circuit utilization over traditional scheduling algorithms, while being within 14% of optimal, at scale.

Gather-Scatter DRAM: In-DRAM Address Translation to Improve the Spatial Locality of Non-unit Strided Accesses

Vivek Seshadri, Thomas Mullins, Amirali Boroumand, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch & Todd C. Mowry

The 48th International Symposium on Microarchitecture (MICRO), Waikiki, Hawaii, USA, December 2015.

Many data structures (e.g., matrices) are typically accessed with multiple access patterns. Depending on the layout of the data structure in physical address space, some access patterns result in non-unit strides. In existing systems, which are optimized to store and access cache lines, non-unit strided accesses exhibit low spatial locality. Therefore, they incur high latency, and waste memory bandwidth and cache space.

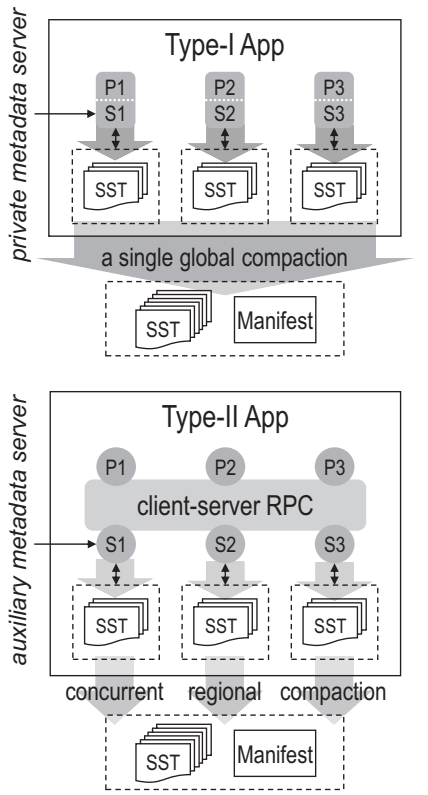
We propose the Gather-Scatter DRAM (GS-DRAM) to address this problem. We observe that a commodity DRAM module contains many chips. Each chip stores a part of every cache line mapped to the module. Our idea is to enable the memory controller to access multiple values that belong to a strided pattern from different chips using a single read/write command. To realize this idea, GS-DRAM first maps the data of each cache line to different chips such that multiple values of a strided access pattern are mapped to different chips. Second, instead of sending a separate address to each chip, GS-DRAM maps each strided pattern to a small pattern ID that is communicated to the module. Based on the pattern ID, each chip independently computes the address of the value to be accessed. The cache line returned by the module contains different values of the strided pattern gathered from different chips. We show that this approach enables GS-DRAM to achieve near-ideal memory bandwidth and cache utilization for many common access patterns. We design an end-to-end system to exploit GS-DRAM. Our evaluations show that 1) for in-memory databases, GS-DRAM obtains the best of the row store and the column store layouts, in terms of both performance and energy, and 2) for matrix-matrix multiplication, GS-DRAM seamlessly enables SIMD optimizations and outperforms the best tiled layout. Our framework is general, and can benefit many modern data-intensive applications.

DeltaFS: Exascale File Systems Scale Better Without Dedicated Servers

Qing Zheng, Kai Ren, Garth Gibson, Bradley W. Settlemeyer & Gary Grider

PDSW2015: 10th Parallel Data Storage Workshop, held in conjunction with SC15, Austin, TX, Nov. 16, 2015.

High performance computing fault tolerance depends on scalable parallel file system performance. For more than a decade scalable bandwidth has



A DeltaFS app runs either with private metadata servers generating a set of overlapping outputs (top), or auxiliary metadata servers holding partitioned outputs (bottom).

been available from the object storage systems that underlie modern parallel file systems, and recently we have seen demonstrations of scalable parallel metadata using dynamic partitioning of the namespace over multiple metadata servers. But even these scalable parallel file systems require significant numbers of dedicated servers, and some workloads still experience bottlenecks. We envision exascale parallel file systems that do not have any dedicated server machines. Instead a parallel job instantiates a file system namespace service in client middleware that operates on only scalable object storage and communicates with other jobs by sharing or publishing namespace snapshots. Experiments shows that our serverless file system design, DeltaFS,

continued on page 21

continued from page 20

performs metadata operations orders of magnitude faster than traditional file system architectures.

The Application Slowdown Model: Quantifying and Controlling the Impact of Inter-Application Interference at Shared Caches and Main Memory

Lavanya Subramanian, Vivek Seshadri, Arnab Ghosh, Samira Khan & Onur Mutlu

The 48th International Symposium on Microarchitecture (MICRO), Waikiki, Hawaii, USA, December 2015.

In a multi-core system, interference at shared resources (such as caches and main memory) slows down applications running on different cores. Accurately estimating the slowdown of each application has several benefits: e.g., it can enable shared resource allocation in a manner that avoids unfair application slowdowns or provides slowdown guarantees. Unfortunately, prior works on estimating slowdowns either lead to inaccurate estimates, do not take into account shared caches, or rely on a priori application knowledge. This severely limits their applicability.

In this work, we propose the Application Slowdown Model (ASM), a new technique that accurately estimates application slowdowns due to interference at both the shared cache and main memory, in the absence of a priori application knowledge. ASM is based on the observation that the performance of each application is strongly correlated with the rate at which the application accesses the shared cache. Thus, ASM reduces the problem of estimating slowdown to that of estimating the shared cache access rate of the application had it been run alone on the system. To estimate this for each application, ASM periodically 1) minimizes interference for the application at the main memory, 2) quantifies the interference the application receives at the shared cache, in an aggregate manner for a large set of requests. Our evaluations across 100 workloads show that ASM has an average slowdown estimation error of only 9.9%, a 2.97x improvement over the best previous mechanism.

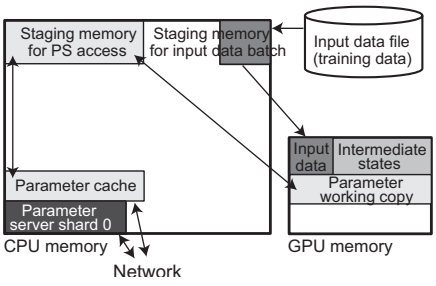
We present several use cases of ASM that leverage its slowdown estimates to improve fairness, performance and provide slowdown guarantees. We provide detailed evaluations of three such use cases: slowdown-aware cache partitioning, slowdown-aware memory bandwidth partitioning and an example scheme to provide soft slowdown guarantees. Our evaluations show that these new schemes perform significantly better than state-of-the-art cache partitioning and memory scheduling schemes.

Scalable Deep Learning on Distributed GPUs with a GPU-specialized Parameter Server

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

Carnegie Mellon University Parallel Data Laboratory Technical Report CMU-PDL-15-107, October 2015.

Large-scale deep learning requires huge computational resources to train a multi-layer neural network. Recent systems propose using 100s to 1000s of machines to train networks with tens of layers and billions of connections. While the computation involved can be done more efficiently on GPUs than on more traditional CPU cores, training such networks on a single GPU is too slow and training on multiple GPUs was also considered unlikely to be effective, due to data movement overheads, GPU stalls, and limited GPU memory. This paper describes a new parameter server, called GeePS, that supports scalable deep learning across GPUs distributed among multiple machines, overcoming these obstacles. We show that GeePS enables a state-of-the-art single-node GPU implementation to scale well, such



Distributed ML on GPUs using a CPU-based parameter server. The right side of the picture is much like a single-GPU configuration. But, a parameter server shard and client-side parameter cache are added to the CPU memory, and the parameter data originally only in the GPU memory is replaced in GPU memory by a local working copy of the parameter data. Parameter updates must be moved between CPU memory and GPU memory, in both directions, which requires an additional application-level staging area since the CPU-based parameter server is unaware of the separate memories.

as to 9.5 times the number of training images processed per second on 16 machines (relative to the original optimized single-node code). Moreover, GeePS achieves the same training throughput with four GPU machines that a state-of-the-art CPU-only system achieves with 108 machines.

Exploiting Inter-Warp Heterogeneity to Improve GPGPU Performance

Rachata Ausavarungnirun, Saugata Ghose, Onur Kayiran, Gabriel H. Loh, Chita R. Das, Mahmut T. Kandemir & Onur Mutlu

Proceedings of the The 24th International Conference on Parallel Architectures and Compilation Techniques (PACT 2015), San Francisco, October 2015.

In a GPU, all threads within a warp execute the same instruction in lockstep. For a memory instruction, this can lead to memory divergence: the memory requests for some threads

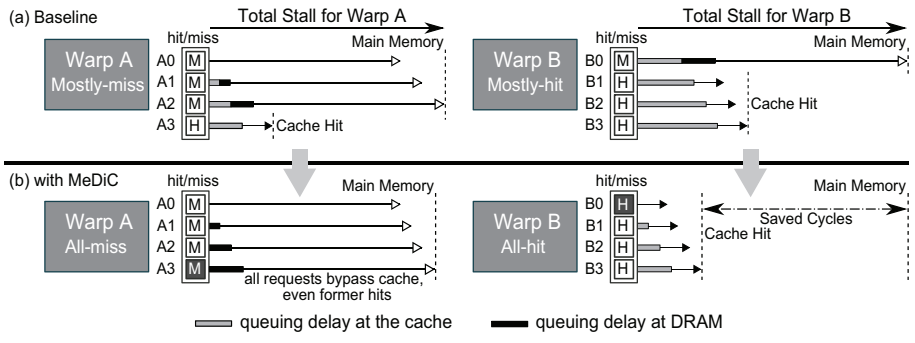
continued on page 22

continued from page 21

are serviced early, while the remaining requests incur long latencies. This divergence stalls the warp, as it cannot execute the next instruction until all requests from the current instruction complete.

In this work, we make three new observations. First, GPGPU warps exhibit heterogeneous memory divergence behavior at the shared cache: some warps have most of their requests hit in the cache (high cache utility), while other warps see most of their request miss (low cache utility). Second, a warp retains the same divergence behavior for long periods of execution. Third, due to high memory level parallelism, requests going to the shared cache can incur queuing delays as large as hundreds of cycles, exacerbating the effects of memory divergence.

We propose a set of techniques, collectively called Memory Divergence Correction (MeDiC), that reduce the negative performance impact of memory divergence and cache queuing. MeDiC uses warp divergence characterization to guide three components: (1) a cache bypassing mechanism that exploits the latency tolerance of low cache utility warps to both alleviate queuing delay and increase the hit rate for high cache utility warps, (2) a cache insertion policy that prevents data from high cache utility warps from being prematurely evicted, and (3) a memory controller that prioritizes the few requests received from high cache utility warps to minimize stall time. We compare MeDiC to four cache management techniques, and find that it delivers an average speedup of 21.8%, and 20.1% higher energy efficiency, over a state-of-the-art GPU cache management mechanism across 15 different GPGPU applications.



(a) Existing inter-warp heterogeneity, (b) exploiting the heterogeneity with MeDiC to improve performance.

Krowd: A Key-Value Store for Crowded Venues

Utsav Drolia, Nathan Mickulicz, Rajeev Gandhi & Priya Narasimhan

10th ACM Workshop on Mobility in the Evolving Internet Architecture (MobiArch), held in Paris, France in September 2015. Best Paper.

Attendees of live events want to capture and share rich content using their mobile devices, during the events. However, the infrastructure at venues that host live events provide poor, low-bandwidth connectivity. Instead of relying on infrastructure provided by the venue, we propose to stand up a temporary "infrastructure" using the very devices that need it, to enable content-sharing with nearby devices. To this end, we developed Krowd, a novel system that provides a key-value store abstraction to applications that share content among local, nearby users. We evaluated Krowd using over 200 hours of real-world traces from sold-out NBA and NHL playoffs and show that it is 50% faster and consumes 50% less bandwidth than alternative systems. We believe that Krowd is the only decentralized and distributed system to provide a key-value store made for neighboring mobile devices and of neighboring mobile devices.

ShardFS vs. IndexFS: Replication vs. Caching Strategies for Distributed Metadata Management in Cloud Storage Systems

Lin Xiao, Kai Ren, Qing Zheng & Garth Gibson

ACM Symposium on Cloud Computing 2015. Aug. 27 - 29, 2015, Kohala Coast, HI.

The rapid growth of cloud storage systems calls for fast and scalable namespace processing. While few commercial file systems offer anything better than federating individually non-scalable namespace servers, a recent academic file system, IndexFS, demonstrates scalable namespace processing based on client caching of directory entries and permissions (directory lookup state) with no per-client state in servers. In this paper we explore explicit replication of directory lookup state in all servers as an alternative to caching this information in all clients. Both eliminate most repeated RPCs to different servers in order to resolve hierarchical permission tests. Our realization for server replicated directory lookup state, ShardFS, employs a novel file system specific hybrid optimistic and pessimistic concurrency control favoring single object transactions over distributed transactions. Our experimentation suggests that if directory lookup state mutation is a fixed fraction of operations (strong scaling for metadata), server replication does not scale as well as client caching, but if directory lookup state mutation is proportional to the number of jobs, not the number of processes per job, (weak scaling for metadata), then server replication can scale more linearly than client caching and provide lower

continued on page 23

continued from page 22

70 percentile response times as well.

Managed Communication and Consistency for Fast Data-Parallel Iterative Analytics

Jinliang Wei, Wei Dai, Aurick Qiao, Qirong Ho, Henggang Cui, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson & Eric P. Xing

ACM Symposium on Cloud Computing 2015. Aug. 27 - 29, 2015, Kohala Coast, HI.

At the core of Machine Learning (ML) analytics applied to Big Data is often an expert-suggested model, whose parameters are refined by iteratively processing a training dataset until convergence. The completion time (i.e. convergence time) and quality of the learned model not only depends on the rate at which the refinements are generated but also the quality of each refinement. While data-parallel ML applications often employ a loose consistency model when updating shared model parameters to maximize parallelism, the accumulated error may seriously impact the quality of refinements and thus delay completion time, a problem that usually gets worse with scale. Although more immediate propagation of updates reduces the accumulated error, this strategy is limited by physical network bandwidth. Additionally, the performance of the widely used stochastic gradient descent (SGD) algorithm is sensitive to initial step size, simply increasing communication without adjusting the step size value accordingly fails to achieve optimal performance.

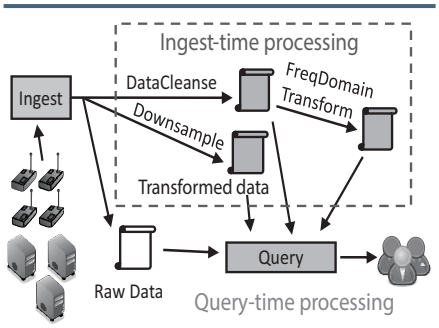
This paper presents Bösen, a system that maximizes the network communication efficiency under a given intermachine network bandwidth budget to minimize accumulated error, while ensuring theoretical convergence guarantees for large-scale data-parallel ML applications. Furthermore, Bösen prioritizes messages that are most significant to algorithm convergence, further enhancing algorithm convergence. Finally, Bösen is the first distributed implementation of the recently presented adaptive revision algorithm, which provides orders of magnitude improvement over a carefully tuned fixed schedule of step size refinements. Experiments on two clusters with up to 1024 cores show that our mechanism significantly improves upon static communication schedules.

Using Data Transformations for Low-latency Time Series Analysis

Henggang Cui, Kimberly Keeton, Indrajit Roy, Krishnamurthy Viswanathan & Gregory R. Ganger

ACM Symposium on Cloud Computing 2015. Aug. 27 - 29, 2015, Kohala Coast, HI.

Time series analysis is commonly used when monitoring data centers, networks, weather, and even human patients. In most cases, the raw time series data is massive, from millions to billions of data points, and yet interactive analyses require low (e.g., subsecond) latency. Aperture transforms raw time series data, during ingest, into compact summarized representations that it can use to efficiently answer queries at runtime. Aperture handles a range of complex queries, from correlating hundreds of lengthy time series to predicting anomalies in the data. Aperture achieves much of its high performance by executing queries on data summaries, while providing a bound on the information lost when transforming data. By doing so, Aperture can reduce query latency as well as the data that needs to be stored and analyzed to answer a query. Our experiments on real data show that Aperture can provide one to four orders of magnitude lower query response time, while incurring only 10% ingest time overhead and less than 20% error in accuracy.



Ingest-time processing and query-time processing. Three transformation outputs are generated from the raw data. FreqDomainTransform is chained after DataCleanse.

Reducing Replication Bandwidth for Distributed Document Databases

Lianghong Xu, Andrew Pavlo, Sudipta Sengupta, Jin Li & Gregory R. Ganger

ACM Symposium on Cloud Computing 2015. Aug. 27 - 29, 2015, Kohala Coast, HI.

With the rise of large-scale, Web-based applications, users are increasingly adopting a new class of document-oriented database management systems (DBMSs) that allow for rapid prototyping while also achieving scalable performance. Like for other distributed storage systems, replication is important for document DBMSs in order to guarantee availability. The network bandwidth required to keep replicas synchronized is expensive and is often a performance bottleneck. As such, there is a strong need to reduce the replication bandwidth, especially for geo-replication scenarios where wide-area network (WAN) bandwidth is limited.

This paper presents a deduplication system called sDedup that reduces the amount of data transferred over the network for replicated document DBMSs. sDedup uses similarity-based deduplication to remove redundancy in replication data by delta encoding against similar documents selected from the entire database. It exploits

continued on page 24

continued from page 23

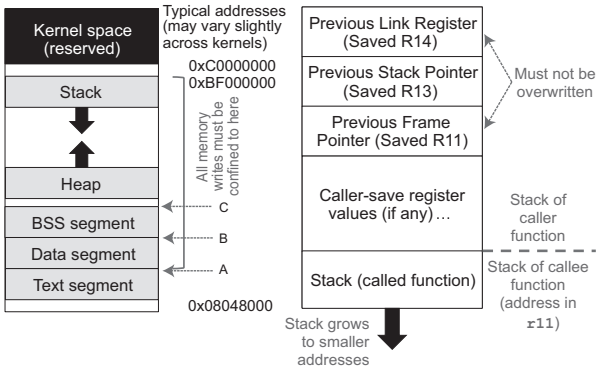
key characteristics of document-oriented workloads, including small item sizes, temporal locality, and the incremental nature of document edits. Our experimental evaluation of sDedup with three real-world datasets shows that it is able to achieve up to 38× reduction in data sent over the network, significantly outperforming traditional chunk-based deduplication techniques while incurring negligible performance overhead.

AUSPICE: Automated Safety Property Verification for Unmodified Executables

Jiaqi Tan, Hui Jun Tay, Rajeev Gandhi & Priya Narasimhan

In 7th Working Conference on Verified Software: Theories, Tools, and Experiments (VSTTE), July 2015.

Verification of machine-code programs using program logic has focused on functional correctness, and proofs have required manually-provided program specifications. Fortunately, the verification of shallow safety properties such as memory and control-flow safety can be easier to automate, but past techniques for automatically verifying machine-code safety have required post-compilation transformations, which can change program behavior. In this work, we automatically verify safety properties for unmodified machine-code programs without requiring user-supplied specifications. We present our novel logic framework, AUSPICE, for automatic safety property verification for unmodified executables, which extends an existing trustworthy Hoare logic for local reasoning, and provides a novel proof tactic for selective composition. We demonstrate our fully automated proof technique on synthetic and realistic programs, and our verification completes in 6 hours for a realistic 533-instruction string search algorithm, demonstrating the feasibility of our approach.



AUSPICE safety property verification.

Architecting to Achieve a Billion Requests Per Second Throughput on a Single Key-Value Store Server Platform

Sheng Li, Hyeontaek Lim, Victor Lee, Jung Ho Ahn, Anuj Kalia, Michael Kaminsky, David G. Andersen, Seongil O, Sukhan Lee & Pradeep Dubey

In Proceedings of the 42nd International Symposium on Computer Architecture (ISCA 2015), Portland, OR, June 2015. Fast-tracked to Transactions on Computer Systems (TOCS).

Distributed in-memory key-value stores (KVSs), such as memcached, have become a critical data serving layer in modern Internet-oriented datacenter infrastructure. Their performance and efficiency directly affect the QoS of web services and the efficiency of datacenters. Traditionally, these systems have had significant overheads from inefficient network processing, OS kernel involvement, and concurrency control. Two recent research thrusts have focused upon improving key-value performance. Hardware-centric research has started to explore specialized platforms including FPGAs for KVSs; results demonstrated an order of magnitude increase in throughput and energy efficiency over stock memcached. Software-centric research revisited the KVS application to address fundamental software bottlenecks and to exploit the full potential of modern commodity hardware; these efforts too showed orders of magnitude improvement over stock memcached.

We aim at architecting high performance and efficient KVS platforms, and start with a rigorous architectural characterization across system stacks over a collection of representative KVS implementations. Our detailed full-system characterization not only identifies the critical hardware/software ingredients for high-performance KVS systems, but also leads to guided optimizations atop a recent design to achieve a recordsetting throughput of 120 million requests per second (MRPS) on a single commodity server. Our implementation delivers 9.2X the performance (RPS) and 2.8X the system energy efficiency (RPS/watt) of the best-published FPGA-based claims. We craft a set of design principles for future platform architectures, and via detailed simulations demonstrate the capability of achieving a billion RPS with a single server constructed following our principles.

Scaling Up Clustered Network Appliances with ScaleBricks

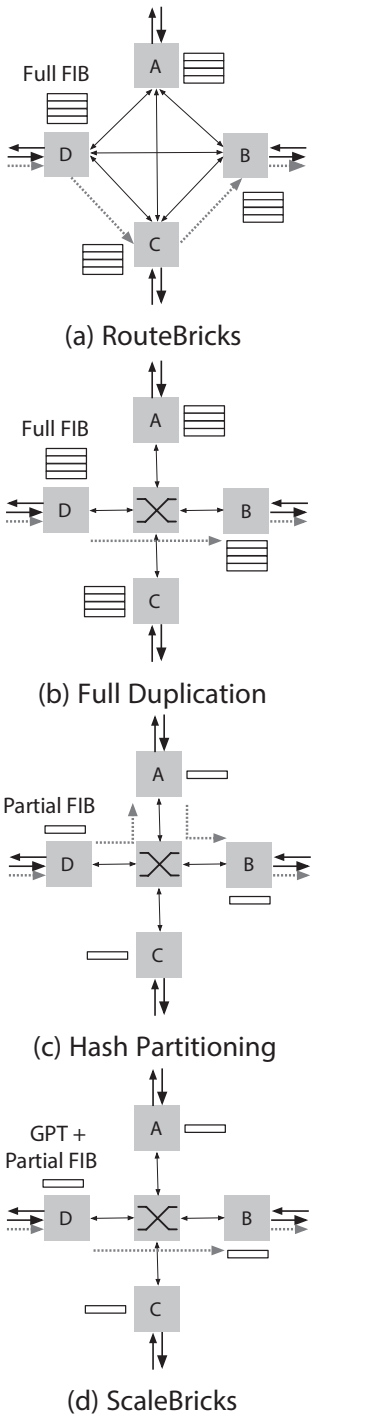
Dong Zhou, Bin Fan, Hyeontaek Lim, David G. Andersen, Michael Kaminsky, Michael Mitzenmacher, Ren Wang & Ajaypal Singh

Proc. ACM SIGCOMM 2015, August 17-21, 2015, London, United Kingdom.

Systematic testing, first demonstrated in small, specialized cases 15 years ago, has matured sufficiently for large-scale

continued on page 25

continued from page 24



Packet forwarding in different FIB architectures.

systems developers to begin to put it into practice. With actual deployment come new, pragmatic challenges to the usefulness of the techniques. In this paper we are concerned with scaling dynamic partial order reduction, a key technique for mitigating the state space explosion problem, to very large clusters. In particular, we present a new approach for distributed dynamic partial order reduction. Unlike previous work, our approach is based on a novel exploration algorithm that 1) enables trading space complexity for parallelism, 2) achieves efficient load-balancing through time-slicing, 3) provides for fault tolerance, which we consider a mandatory aspect of scalability, 4) scales to more than a thousand parallel workers, and 5) is guaranteed to avoid redundant exploration of overlapping portions of the state space.

Reducing Latency via Redundant Requests: Exact Analysis

Kristen Gardner, Sam Zbarsky, Sherwin Doroudi, Mor Harchol-Balter, Esa Hyytia & Alan Scheller-Wolf

Proceedings of ACM Sigmetrics/Performance 2015 Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 15), Portland, OR. June 2015.

Recent computer systems research has proposed using redundant requests to reduce latency. The idea is to run a request on multiple servers and wait for the first completion (discarding all remaining copies of the request). However there is no exact analysis of systems with redundancy.

This paper presents the first exact analysis of systems with redundancy. We allow for any number of classes of redundant requests, any number of classes of non-redundant requests, any degree of redundancy, and any number of heterogeneous servers. In all cases we derive the limiting distribution on the state of the system.

In small (two or three server) systems, we derive simple forms for the distribution of response time of both the redundant classes and non-redundant classes, and we quantify the "gain" to redundant classes and "pain" to non-redundant classes caused by redundancy. We find some surprising results. First, the response time of a fully redundant class follows a simple Exponential distribution and that of the non-redundant class follows a Generalized Hyper-exponential. Second, fully redundant classes are "immune" to any pain caused by other classes becoming redundant.

We also compare redundancy with other approaches for reducing latency, such as optimal probabilistic splitting of a class among servers (Opt-Split) and Join-the-Shortest-Queue (JSQ) routing of a class. We find that, in many cases, redundancy outperforms JSQ and Opt-Split with respect to overall response time, making it an attractive solution.

Caveat-Scriptor: Write Anywhere Shingled Disks

Saurabh Kadekodi, Swapnil Pimpale & Garth Gibson

Proc. Of the Seventh USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage'15), Santa Clara, CA, July 2015.

The increasing ubiquity of NAND flash storage is forcing magnetic disks to accelerate the rate at which they lower price per stored bit. Magnetic recording technologists have begun to pack tracks so closely that writing one track cannot avoid disturbing the information stored in adjacent tracks [13]. Specifically, the downstream track will be at least partially overwritten, or shingled by each write, as shown in Figure 1, and the upstream track will tolerate only a limited number of adjacent writes. Some data that was stored in the downstream track will be lost, forcing firmware or software to ensure that there was no data in those locations that might be read in the future.

continued on page 26

continued from page 25

Page Overlays: An Enhanced Virtual Memory Framework to Enable Fine-grained Memory Management

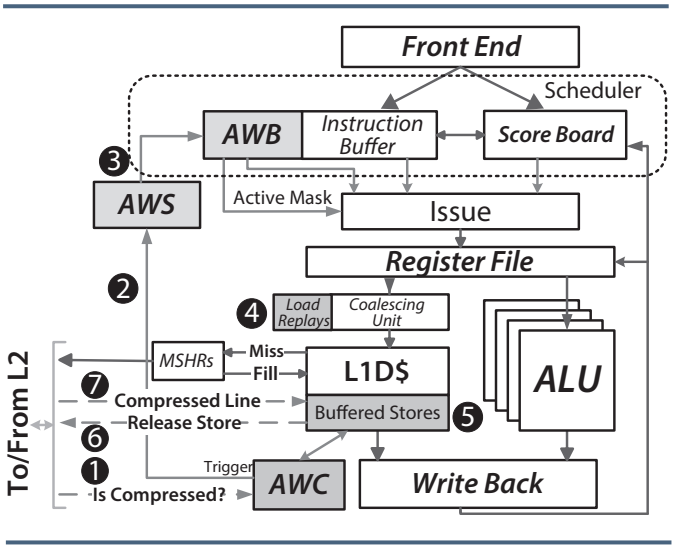
Vivek Seshadri, Gennady Pekhimenko, Olatunji Ruwase, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, Todd C. Mowry & Trishul Chilimbi

Proceedings of the 42nd International Symposium on Computer Architecture (ISCA), Portland, OR, June 2015.

Many recent works propose mechanisms demonstrating the potential advantages of managing memory at a fine (e.g., cache line) granularity—e.g., fine-grained deduplication and fine-grained memory protection. Unfortunately, existing virtual memory systems track memory at a larger granularity (e.g., 4 KB pages), inhibiting efficient implementation of such techniques. Simply reducing the page size results in an unacceptable increase in page table overhead and TLB pressure.

We propose a new virtual memory framework that enables efficient implementation of a variety of fine-grained memory management techniques. In our framework, each virtual page can be mapped to a structure called a page overlay, in addition to a regular physical page. An overlay contains a subset of cache lines from the virtual page. Cache lines that are present in the overlay are accessed from there and all other cache lines are accessed from the regular physical page. Our page-overlay framework enables cache-line-granularity memory management without significantly altering the existing virtual memory framework or introducing high overheads.

We show that our framework can enable simple and efficient implementations of seven memory management techniques, each of which has a wide variety of applications. We quantitatively evaluate the potential benefits of two of these techniques: overlay-on-write and sparse-data-structure computation. Our evaluations show that overlay-on-write, when applied to fork, can improve performance by 15% and reduce memory capacity requirements by 53% on average compared to traditional copy-on-write. For sparse data computation, our framework can outperform a state-of-the-art software-based sparse representation on a number of real-world sparse matrices. Our framework is general, powerful, and effective in enabling fine-grained memory management at low cost.



Walkthrough of CABA-based Compression. (See the paper for a description of the workflow).

A Case for Core-Assisted Bottleneck Acceleration in GPUs: Enabling Efficient Data Compression

Nandita Vijaykumar, Gennady Pekhimenko, Adwait Jog, Abhishek Bhowmick, Rachata Ausavarungnirun, Chita Das, Mahmut Kandemir, Todd C. Mowry & Onur Mutlu

The 42nd International Symposium on Computer Architecture (ISCA), Portland, OR, June 2015.

Modern Graphics Processing Units (GPUs) are well provisioned to support the concurrent execution of thousands of threads. Unfortunately, different bottlenecks during execution and heterogeneous application requirements create imbalances in utilization of resources in the cores. For example, when a GPU is bottlenecked by the available off-chip memory bandwidth, its computational resources are often overwhelmingly idle, waiting for data from memory to arrive.

This paper introduces the Core-Assisted Bottleneck Acceleration (CABA) framework that employs idle on-chip resources to alleviate different bottlenecks in GPU execution. CABA provides flexible mechanisms to automatically generate "assist warps" that execute on GPU cores to perform specific tasks that can improve GPU performance and efficiency.

CABA enables the use of idle computational units and pipelines to alleviate the memory bandwidth bottleneck, e.g., by using assist warps to perform data compression to transfer less data from memory. Conversely, the same framework can be employed to handle cases where the GPU is bottlenecked by the available computational units, in which case the memory pipelines are idle and can be used by CABA to speed up computation, e.g., by performing memoization using assist warps.

We provide a comprehensive design and evaluation of CABA to perform effective and flexible data compression in the GPU memory hierarchy to alleviate the memory bandwidth bottleneck. Our extensive evaluations show that CABA, when used to implement data compression, provides an average performance improvement of 41.7% (as high as 2.6X) across a variety of memory-bandwidth-sensitive GPGPU applications.

continued on page 27

continued from page 26

PocketTrend: Timely Identification and Delivery of Trending Search Content to Mobile Users

Gennady Pekhimenko, Dimitrios Lymberopoulos, Oriana Riva, Karin Strauss & Doug Burger

Proceedings of the 24th International World Wide Web Conference (WWW), Florence, Italy, May 2015.

Trending search topics cause unpredictable query load spikes that hurt the end-user search experience, particularly the mobile one, by introducing longer delays. To understand how trending search topics are formed and evolve over time, we analyze 21 million queries submitted during periods where popular events caused search query volume spikes. Based on our findings, we design and evaluate PocketTrend, a system that automatically detects trending topics in real time, identifies the search content associated to the topics, and then intelligently pushes this content to users in a timely manner. In that way, PocketTrend enables a client-side search engine that can instantly answer user queries related to trending events, while at the same time reducing the impact of these trends on the datacenter workload. Our results, using real mobile search logs, show that in the presence of a trending event, up to 13–17% of the overall search traffic can be eliminated from the datacenter, with as many as 19% of all users benefiting from PocketTrend.

BIG-LEARNING SYSTEMS FOR BIG DATA

continued from page 13

see "Solving the Straggler Problem for Iterative Convergent Parallel ML" [7].

Scheduled Model Parallelism. Although data-parallel implementations are the norm, an alternative and complementary strategy called *model parallel* is sometimes more effective. This strategy partitions model parameters for non-shared parallel access and updates, rather than having all threads access all parameters; so refinement of model parameters is partitioned among workers instead of the training data. Model parallelism can deal better with parameters that are dependent in ways that make concurrent adjustments induce too much error and with parameters that converge at very different rates. But, traditionally, model parallel implementations have been more complex to develop and less able to adapt to differing resource availability. Our STRADS system demonstrates a new approach, called scheduled model parallelism (SchMP), and shows that it can improve ML algorithm convergence speed by efficiently scheduling parameter updates, taking into account parameter dependencies and uneven convergence. For more information, see "STRADS: A Distributed Framework for Scheduled Model Parallel Machine Learning" [8].

References

[1] More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server. Qirong Ho, James Cipar, Henggang Cui, Jin Kyu Kim, Seunghak Lee, Phillip B. Gibbons, Garth A. Gibson, Gregory R. Ganger, Eric P. Xing. NIPS '13, Dec. 2013, Lake Tahoe, NV.

[2] Scaling Distributed Machine Learning with the Parameter Server Mu Li, Dave Andersen, Alex Smola, Junwoo Park, Amr Ahmed, Vanja Josifovski, James Long, Eugene Shekita, Bor-Yiing Su. OSDI'14.

[3] Exploiting Bounded Staleness to Speed up Big Data Analytics. Henggang Cui, James Cipar, Qirong Ho, Jin Kyu Kim, Seunghak Lee, Abhimanu Kumar Jinliang Wei, Wei Dai, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson, Eric P. Xing. ATC'14, Philadelphia, PA.

[4] Exploiting Iterative-ness for Parallel ML Computations. Henggang Cui, Alexey Tumanov, Jinliang Wei, Lianghong Xu, Wei Dai, Jesse Haber-Kucharsky, Qirong Ho, Greg R. Ganger, Phil B. Gibbons, Garth A. Gibson, Eric P. Xing. SoCC'14, Seattle, WA, Nov. 2014.

[5] Managed Communication and Consistency for Fast Data-Parallel Iterative Analytics. Jinliang Wei, Wei Dai, Aurick Qiao, Qirong Ho, Henggang Cui, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson, Eric P. Xing. SoCC'15, Kohala Coast, HI.

[6] GeePS: Scalable Deep Learning on Distributed GPUs with a GPU-Specialized Parameter Server. Henggang Cui, Hao Zhang, Gregory R. Ganger, Phillip B. Gibbons, and Eric P. Xing. EuroSys'16, London, UK.

[7] Solving the Straggler Problem for Iterative Convergent Parallel ML. Aaron Harlap, Henggang Cui, Wei Dai, Jinliang Wei, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson, Eric P. Xing. Carnegie Mellon University Parallel Data Laboratory Technical Report CMU-PDL-15-102, April 2015.

[8] STRADS: A Distributed Framework for Scheduled Model Parallel Machine Learning. Jin Kyu Kim, Qirong Ho, Seunghak Lee, Xun Zheng, Wei Dai, Garth A. Gibson, Eric P. Xing. EuroSys'16, London, UK.

YEAR IN REVIEW

continued from page 4

- Utsav Drolia and his co-authors received the Best-Paper Award at MobiArch'15 in Paris, France for the paper "Krowd: A Key-Value Store for Crowded Venues."
- Wolfgang Richter successfully defended his PhD research on "Agentless Cloud-wide Monitoring of Virtual Disk State."
- Lorrie Cranor, Mor Harchol-Balter and Andy Pavlo each received Google Faculty Research Awards.
- Andy Pavlo and Mor Harchol-Balter each received a Facebook Faculty Award.

August 2015

- Jinliang Wei and co-authors received a Best Paper award at SoCC'15, held on the Kohala Coast, HI for their paper "Managed Communication and Consistency for Fast Data-Parallel Iterative Analytics."
- Also presented at SoCC'15: "Using Data Transformations for Low-latency Time Series Analysis" (Henggang Cui) and "Reducing Replication Bandwidth for Distributed Document Databases" (Lianghong Xu).

July 2015

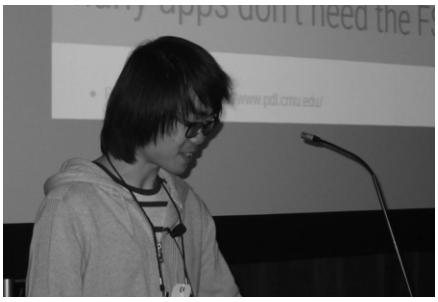
- Hyeontaek Lim successfully defended his PhD dissertation "Resource-Efficient Data-Intensive System Designs for High Performance and Capacity" and transitioned to postdoc status with the PDL.

June 2015

- Junchen Jiang proposed his thesis research "Better End-to-End Adaptation Using Centralized Predictive Control."
- Dave Anderson's group's paper "Architecting to Achieve a Billion Requests Per Second Throughput on a Single Key-Value Store Server Platform" was presented at ISCA'15 in Portland, OR and fast-tracked to TOCS!

May 2015

- Alexey Tumanov received ECE's Graduate Student Teaching Assistant Award.
- Gennady Pekhimenko received an NVIDIA graduate fellowship.
- Alexey Tumanov proposed his PhD research on "Scheduling with Space-Time Soft Constraints in Heterogeneous Cloud Datacenters."
- Gennady Pekhimenko presented "PocketTrend: Timely Identification and Delivery of Trending Search Content to Mobile Users" at WWW'15 in Florence, Italy, May 2015.
- 17th annual PDL Spring Visit Day.



Qing Zheng presents his research on "DeltaFS: Scalable File System For Future Exascale Data Centers" at the 2015 PDL Retreat.

2015 PDL Workshop and Retreat.