Intel Science & Technology Center for Cloud Computing

ISTC-CC Update

August 2014

www.istc-cc.cmu.edu

Table of Contents

Carnegie Mellon University

Georgia Tech

intel

PRINCETON UNIVERSITY

UC Berkeley

UNIVERSITY of WASHINGTON

ISTC-CC Research Overview

Cloud computing has become a source of enormous buzz and excitement, promising great reductions in the effort of establishing new applications and services, increases in the efficiency of operating them, and improvements in the ability to share data and services. Indeed, we believe that cloud computing has a bright future and envision a future in which nearly all storage and computing is done via cloud computing resources. But, realizing the promise of cloud computing will require an enormous amount of research and development across a broad array of topics.

ISTC-CC was established to address a critical part of the needed advancement: underlying cloud infrastructure technologies to serve as a robust, efficient foundation for cloud applications. The ISTC-CC research agenda is organized into four inter-related research "pillars" (themes) architected to create a strong foundation for cloud computing of the future:

The four ISTC-CC pillars provide a strong foundation for cloud computing of the future, delivering cloud's promised benefits to the broad collection of applications and services that will rely on it.

Pillar 1: Specialization

Driving greater efficiency is a significant global challenge for cloud datacenters. Current approaches to cloud deployment, especially for increasingly popular private clouds, follow traditional data center practices of identifying a single server architecture and avoiding heterogeneity as much as possible. IT staff have long followed such practices to reduce administration complexity—homogeneity yields uniformity, simplifying many aspects of maintenance, such as load balancing, inventory, diagnosis, repair, and so on. Current best practice tries to find a configuration that is suitable for all potential uses of a given infrastructure.

Unfortunately, there is no single server configuration that is best, or close to best, for all applications. Some applications are computation-heavy, needing powerful CPUs

continued on pg. 30

Message from the PIs

Hello from ISTC-CC headquarters. This ISTC-CC Newsletter, our third, includes our general ISTC-CC research overview, news and happenings from the last 11 months, and abstracts of our many publications. While we can't recap all that has happened in this introductory note, we do want to highlight a few things.

Greg Ganger, CMU

Phil Gibbons, Intel

First, a bit of bragging. As might be expected, given the exceptional team of Intel and academic researchers working together, ISTC-CC has been extremely successful. ISTC-CC projects have made, and continue to make, major impact within Intel and in the greater community, both in underlying ideas and technologies and in open source software systems. We're very pleased that Intel leadership continues to embrace ISTC-CC, including officially renewing the center for another two years. The continued commitment of all the participants promises great things to come, as some of the most promising project results are brought together into bigger capstone efforts.

What makes ISTC-CC work so well is that the team is more than the sum of its parts — the individuals are stellar, but they are also great collaborators. So many of ISTC-CC's big wins come from teams within and across the 6 participating institutions. Indeed, many of the technical papers and software artifacts involve researchers from multiple institutions... and a third of them have Intel co-authors. It's a lot of fun working with folks like these!

Speaking of software artifacts, we want to draw attention to a recent addition to the ISTC-CC website: the ISTC-CC software page. To make them easier to find, we are collecting links to our software releases and open source development efforts. We continue to add to it, as we try to make our efforts ever more useful to Intel, ISTC-CC and the broader community.

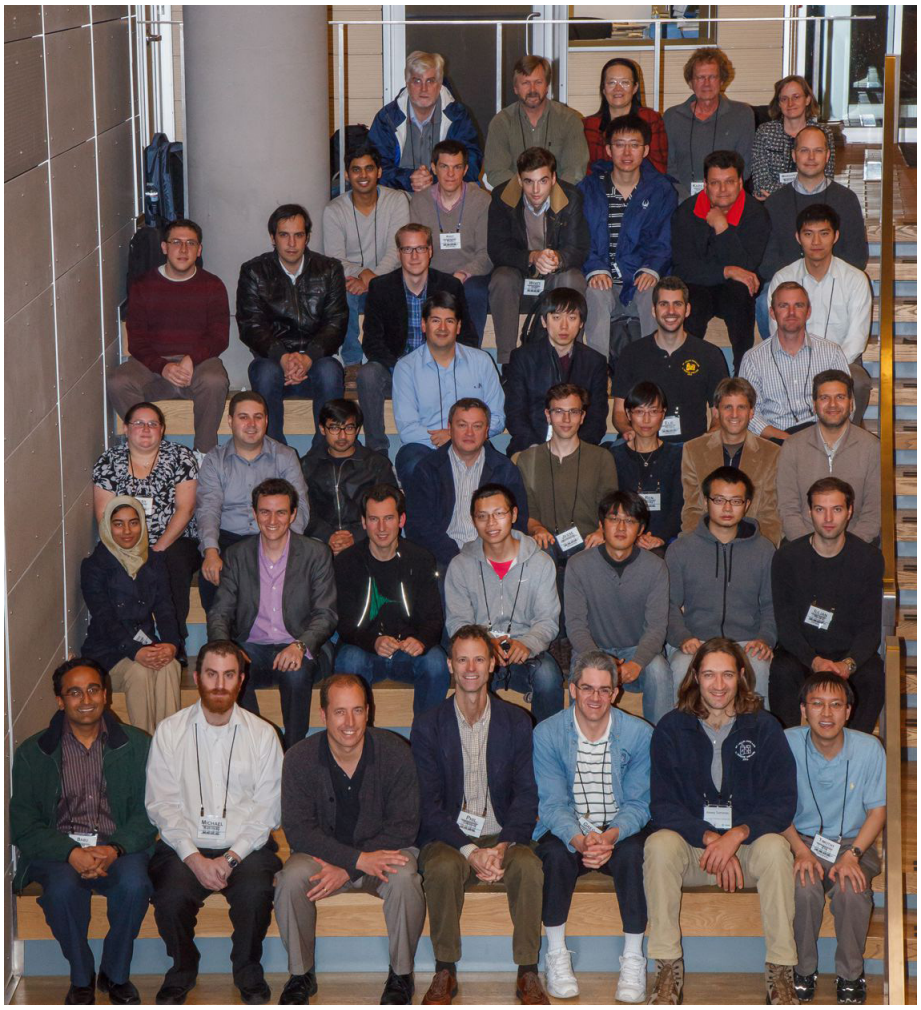
As described in the ISTC-CC overview article, we continue to describe the overall ISTC-CC agenda in terms of four inter-related "pillars" — specialization, automation, big data, to the edge — designed to enable cloud computing infrastructures that provide a strong foundation for future cloud computing. (We're guiltily proud of the pillar metaphor.) But, the categorization is for agenda presentation purposes only, as the activities increasingly span pillars, such as scheduling (automation) of multiple data-intensive frameworks (big data) across heterogeneous (specialized) cluster resources. Indeed, our capstone efforts will naturally involve activities from different areas.

One area where ISTC-CC impact has been huge is something we call "big learning systems": (new) frameworks for supporting efficient Big Data analytics based on advanced machine learning (ML) algorithms. In particular, ISTC-CC's GraphLab and Spark have become very popular open source systems in addition to changing mindsets on the right way to enable ML on Big Data. Lots of energy and entire software ecosystems are growing up around both, including adoption and contributions by Intel. ISTC-CC continues to develop a range of more effective and natural abstractions for different types of non-trivial ML tasks and designing frameworks to enable them, such as consistency models based on bounded staleness and parallelization architectures based on what we call "parameter servers" (servers for widely

continued on pg. 32

Third Annual ISTC-CC Retreat a Success!

The ISTC-CC held its third annual retreat in Pittsburgh on November 7-8, 2013. The 106 attendees included faculty and students from Carnegie Mellon, Georgia Tech, Princeton, UC Berkeley & Washington, as well as 21 Intel employees. The agenda featured keynotes by Rich Uhlig, Pradeep Dubey, and Myles Wilde of Intel, 13 research talks by faculty and students from all five Universities, 4 BoF sessions, and 44 posters. By all accounts, the retreat was a big success: great interactions, lots of connections made, new insights, idea inspiration, and generally superb energy! The retreat was followed by the Board of Advisors meeting where Greg Ganger and Phil Gibbons presented the ISTC-CC's plans for years 3-5, and an additional meeting for Intel stakeholders. These meetings provided considerable positive feedback, as well as good suggestions. In early December we learned that the ISTC-CC was renewed for years 4-5 at the $2M/year level. Full details on the retreat can be found on the ISTC-CC website. Note that the fourth ISTC-CC Retreat is scheduled for September 4-5, 2014 at Intel's Jones Farm site in Hillsboro, Oregon.



Group photo — third annual ISTC-CC Retreat, November 2013.

ISTC-CC Personnel

Leadership

Greg Ganger, Academic PI
Phil Gibbons, Intel PI
Executive Sponsor: Rich Uhlig, Intel
Managing Sponsor: Scott Hahn, Intel
Program Director: Jeff Parkhurst, Intel
Board of Advisors:
Randy Bryant, CMU
Jeff Chase, Duke
Balint Fleisher, Intel
Frans Kaashoek, MIT
Pradeep Khosla, UC San Diego
Jason Waxman, Intel

Faculty

David Andersen, CMU
Guy Blelloch, CMU
Greg Eisenhauer, GA Tech
Mike Freedman, Princeton
Greg Ganger, CMU
Ada Gavrilovska, GA Tech
Phillip Gibbons, Intel
Garth Gibson, CMU
Carlos Guestrin, U. Washington
Mor Harchol-Balter, CMU
Anthony Joseph, Berkeley
Randy Katz, Berkeley
Ling Liu, GA Tech
Michael Kaminsky, Intel
Mike Kozuch, Intel
Margaret Martonosi, Princeton
Todd Mowry, CMU
Onur Mutlu, CMU
Priya Narasimhan, CMU
Padmanabhan (Babu) Pillai, Intel
Calton Pu, GA Tech
Mahadev (Satya) Satyanarayanan, CMU
Karsten Schwan, GA Tech
Dan Siewiorek, CMU
Alex Smola, CMU
Ion Stoica, Berkeley
Matthew Wolf, GA Tech
Sudhakar Yalamanchili, GA Tech
Eric Xing, CMU

Staff

Joan Digney, Editor/Web, CMU
Jennifer Gabig, ISTC Admin. Manager, CMU

Students / Post-Docs

Yoshihisa Abe, CMU
Sameer Agarwal, Berkeley
Rachata Ausavarungnirun, CMU
Ben Blum, CMU
Kevin Kai-Wei Chang, CMU
Zhuo Chen, CMU
Anthony Chivetta, CMU
Henggang Cui, CMU
Wei Dai, CMU
Kristen Gardner, CMU
Ali Ghodsi, Berkeley
Michelle Goodstein, CMU
Joseph Gonzalez, CMU
Samantha Gottlieb, CMU
Haijie Gu, CMU
Mehgana Gupta, GA Tech
Kiryong Ha, CMU
Jesse Haber-Kucharsky, CMU
Liting Hu, GA Tech
Wenlu Hu, CMU
Lu Jiang, CMU
Tyler Johnson, Washington
Anuj Kalia, CMU
Sudarsun Kannan, GA Tech
Mike Kasick, CMU
Deby Katz, CMU
Samira Khan, CMU
Jin Kyu Kim, CMU
Yoongu Kim, CMU
Andy Konwinski, Berkeley
Elie Krevat, CMU
Abhimanu Kumar, CMU
Guatam Kumar, Berkeley
Aapo Kyrola, CMU
Seunghak Lee, CMU
Mu Li, CMU
Hyeontaek Lim, CMU
Daniel Lustig, Princeton
Justin Meza, CMU
Ishan Misra, CMU
Jun Woo Park, CMU
Gennady Pekhimenko, CMU
Ram Raghunathan, CMU
Kai Ren, CMU
Wolfgang Richter, CMU
Vivek Seshadri, CMU
Julian Shun, CMU
Lavanya Subramanian, CMU
Logan Stafman, Princeton
Jiaqi Tan, CMU
Brandon Taylor, CMU
Caroline Trippel, Princeton
Alexey Tumanov, CMU
Jinliang Wei, CMU
Haicheng Wu, GA Tech
Jin Xin, Princeton
Lianghong Xu, CMU
Hobin Yoon, GA Tech
David Zats, Berkeley
Huanchen Zhang, CMU
Xun Zheng, CMU
Dong Zhou, CMU
Timothy Zhu, CMU

The ISTC-CC Update

The Newsletter for the Intel Science and Technology Center for Cloud Computing

Carnegie Mellon University
ISTC-CC
CIC 4th Floor
4720 Forbes Avenue
Pittsburgh, PA 15213
T (412) 268-2476

EDITOR

Joan Digney

The ISTC-CC Update provides an update on ISTC-CC activities to increase awareness in the research community.

THE ISTC-CC LOGO

ISTC logo embodies its mission, having four inter-related research pillars (themes) architected to create a strong foundation for cloud computing of the future.

The research agenda of the ISTC-CC is composed of the following four themes.

Specialization: Explores specialization as a primary means for order of magnitude improvements in efficiency (e.g., energy), including use of emerging technologies like non-volatile memory and specialized cores.

Automation: Addresses cloud's particular automation challenges, focusing on order of magnitude efficiency gains from smart resource allocation/scheduling and greatly improved problem diagnosis capabilities.

Big Data: Addresses the critical need for cloud computing to extend beyond traditional big data usage (primarily, search) to efficiently and effectively support Big Data analytics, including the continuous ingest, integration, and exploitation of live data feeds (e.g., video or social media).

To the Edge: Explores new frameworks for edge/cloud cooperation that can efficiently and effectively exploit billions of context-aware clients and enable cloud-assisted client applications whose execution spans client devices, edge-local cloud resources, and core cloud resources.

Year in Review

This section lists a sampling of significant ISTC-CC occurrences in the past 11 months.

Ion Stoica (UC Berkeley) talks about his work on "Discretized Streams: Fault-Tolerant Streaming Computation at Scale" at the 2013 Retreat.

2013 Quarter 4

» Anshul Gandhi (ISTC-CC alum) won the SPEC Dissertation award for his PhD thesis, titled, "Dynamic Server Provisioning for Data Center Power Management."

» Mor Harchol-Balter (CMU) gave an invited talk on "Dynamic Power Management of Data Centers: Theory and Practice" at DIMACS Working Group on Algorithms for Green Data Storage, December 2013.

» Michael J. Freedman (Princeton) presented "Multi-tenant Resource Allocation for Shared Cloud Storage" at the New Results in Networking Research 2013, Microsoft Research, Redmond, WA, December 2013.

» Onur Mutlu (CMU) gave a keynote talk at the Industry-Academia Partnership Stanford Cloud Workshop on "Rethinking Memory System Design for Data-Intensive Computing," Mountain View, CA, December 2013.

» Amplifying funding was received by a proposal by Mahadev Satyanarayanan (PI) and co-PIs, Dan Siewiorek, Jason Hong, and Asim Smailagic (CMU), entitled, "QuiltView: Glass-Sourced Video for Google Maps Queries" from Google.

» Phil Gibbons (Intel Labs) gave a distinguished lecture at EPFL on "The Intel Science and Technology Center for Cloud Computing," Lausanne, Switzerland, December 2013.

» Phil Gibbons (Intel Labs), Garth Gibson (CMU), and Sudhakar Yalamanchili (GA Tech) were elected IEEE Fellows.

» A collaborative effort between personnel at CMU, Georgia Tech and Intel resulted in the Compressed Buffer Tree (CBT) open source code release.

» Dan Siewiorek (CMU) received an ACM/IEEE Design Automation Conference's Second Decade (1974-1983) Award Top 10 Author Award, and a Prolific Author Award.

» Joseph Gonzales (CMU) was nominated for ACM Outstanding Dissertation for his work on "Parallel and Distributed Systems for Probabilistic Reasoning."

» Dong Zhou, Bin Fan, Hyeontaek Lim (CMU grad students), Michael Kaminsky (Intel Labs), and David Andersen (CMU), were nominated for Best Paper at CoNEXT'13 for their paper on fast switching based on cuckoo hashing.

» Michael Kaminsky and Babu Pillai (Intel Labs) organized a highly successful SOSP'13 conference.

» Karsten Schwan (GA Tech) was PC co-chair for ACM Middleware in Beijing, December 2013.

2014 Quarter 1

» Greg Ganger received the 2014 Steven J. Fenves Award.

» Margaret Martonosi (Princeton) presented "Power-Aware Computing: Then, Now and into the Future" at University of Wisconsin and at the University of Ghent, Belgium, March 2014.

» Sudha Yalamanchili (GA Tech) presented "Red Fox: An Execution Environment for Relational Queries Processing on GPUs" at the GPU Technology Conference, March 2014.

» Phil Gibbons (IL) served as chair for the "Big Data, Data Management and Analytics" track of the 34th International Conference on Distributed Computing Systems (ICDCS'14). Garth Gibson (CMU), Michael Kozuch (IL), Ling Liu (GA Tech), Priya Narasimhan (CMU), Calton Pu (GA Tech), and Karsten Schwan (GA Tech) served on the ICDCS'14 PC.

» Mor Harchol-Balter (CMU) joined 3 technical program committees: ACM SIGMETRICS, IFIP PERFORMANCE, and ECQT (European Conference on Queueing Theory).

Year in Review

» Phil Gibbons (IL) joined the PC for SOCC'14, and reviewed papers as a PC member for SPAA'14.
» Mor Harchol-Balter's (CMU) textbook, "Performance Modeling & Design of Computer Systems" is being used at Columbia University and Washington University, St. Louis in this semester's classes.
» Berkeley had two open source code releases: Tachyon 0.4.1, and Spark 0.9.0; CMU had two open source code releases: Networked key-value cache MICA and Concurrent cuckoo hash table. For more info on ISTC-CC's code releases, please see www.istc-cc.cmu.edu/research/ossr/.
» Justin Meza and Lavanya Subramanian (both CMU) were each awarded a Bertucci Fellowship to continue their PhD studies.
» Several CMU graduate students under the direction of M. Satyanarayanan won the Best Demo Award at HotMobile'14 for their work on "QuiltView: Glass-Sourced Video for Google Maps Queries."

2014 Quarter 2

» Mor Harchol-Balter (CMU) received two awards as a result of her teaching (to 400 freshmen) of class 21-127 Proof Concepts: (i) 2014 CMU Mudge House Dinner with the Deans Honorary Event for Influential Teachers, and (ii) 2014 Apple Pie with Alpha Chi Honorary Event for CMU Faculty with Impact on Students.
» Onur Mutlu (CMU) and his co-author's paper "Bounding Memory Interference Delay in COTS-based Multi-Core Systems" won best paper award at RTAS'14.

Wenlu Hu (CMU) takes 30 seconds to preview her poster on "QuiltView: a Crowd-Sourced Video Response System" at the 2013 ISTC-CC Retreat.

» Greg Ganger (CMU) gave a keynote talk on "Scheduling Heterogeneous Resources in Cloud Datacenters," at the IAP Cloud Workshop at CMU, April 2014.
» Mor Harchol-Balter (CMU) presented "Value Driven Load Balancing" at the MSR-CMU Mind Swap in NYC, April 2014.
» Eric Xing (CMU) presented "Petuum: A New Platform for Cloud-based Machine Learning to Efficiently Solve Big Data Problems" at the Industry-Academia Partnership (IAP) Cloud Workshop at CMU, April 2014.
» Onur Mutlu (CMU) and his grad student Yoongu Kim published an invited book chapter on Memory Systems in Computing Handbook, Third Edition: Computer Science and Software Engineering, CRC Press, April 2014.
» Samira Khan's (CMU) paper, "Improving Cache Performance by Exploiting Read-Write Disparity" was presented in a Best Paper Session at HPCA'14.
» Dan Siewiorek (CMU) gave two keynote talks on "Generation Smart Phone: The Smartphone's Role as Constant Companion, Helper, Coach, and Guardian Has Only Just Begun," at the Harbin Institute of Technology, Harbin China, and at the Chinese University of Hong Kong, China, May 2014.
» Dan Siewiorek (CMU) gave a keynote talk on "Overview of Quality of Life Technology Engineering Research Center" at the IEEE Big Data 2014 Shenzhen Satellite Session, Shenzhen, China, May 2014.
» Onur Mutlu (CMU) gave three keynote talks on "Rethinking Memory/Storage System Design for Data-Intensive Computing" at (i) IAP Cloud Workshop at CMU, April 2014, (ii) Huawei Strategy and Technology Workshop, May 2014, and (iii) Green, Pervasive, Cloud Computing Conference in Wuhan, China, May 2014.
» Ion Stoica (UC Berkeley) gave a keynote entitled "Taming Big Data with Berkeley Data Analytics Stack (BDAS)" at the 14th IEEE/ACM Int'l Symposium on Cluster, Cloud and Grid Computing (CCGrid'14), May 2014.
» Kevin Chang (CMU grad student) received the Intel Foundation/SRC-CEA Graduate Fellowship.
» Wolfgang Richter (CMU) and his co-authors won the International Conference on Cloud Engineering (IC2E) Best Paper Award for "Agentless Cloud-wide Streaming of Guest File System Updates."
» Garth Gibson (CMU) served as Program Co-Chair for the Usenix Annual Technical Conference (ATC'14).
» Michael Kozuch (Intel Labs) served as Program Co-Chair for HotCloud'14.
» Mor Harchol-Balter's (CMU) textbook, "Performance Modeling and Design of Computer Systems" was adopted this past semester as a textbook for classes at the following universities: Columbia University, Washington University in St. Louis, University of Beirut (Lebanon), University of South Carolina, and McMaster University (Canada).
» Bin Fan, Dave Andersen, and Michael Kaminsky released the MemC3 code (MemC3 was an NSDI'13 paper): https://github.com/efficient/memc3. MemC3 is an in-memory key-value cache, derived from Memcached but improved with memory-efficient and concurrent data structures. MemC3 applies multi-reader concurrent cuckoo hashing as its key-value index and CLOCK-replacement algorithm as cache eviction policy. As a result, MemC3 scales better, runs faster, and uses less memory.
» Onur Mutlu (CMU) gave three keynote talks on "Rethinking Memory/Storage System Design for Data-Intensive Computing" at IAP Cloud Workshop at CMU, April 2014, at the Huawei Strategy and Technology Workshop, May 2014, and at Green, Pervasive, Cloud Computing Conference in Wuhan, China, May 2014.
» Mor Harchol-Balter (CMU) presented "Dynamic Power Management of Data Centers" at the Industry-Academia Partnership (IAP) Cloud Workshop at CMU, April 2014 and at the LCCC Workshop in Cloud Control, Lund University, Sweden, May 2014.
» Dan Siewiorek (CMU) presented "Generation Smart Phone: The Smartphone's Role as Constant Companion, Helper, Coach, and

continued on pg. 32

ISTC-CC News

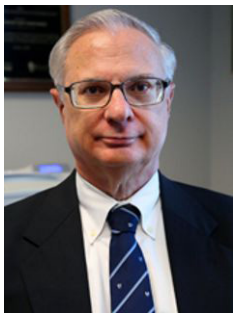
August 24, 2014

Best Paper Award!

Aaron Li, Amr Ahmed, Sujith Ravi, Alex Smola have won the KDD 2014 best paper prize for their work on "Reducing the sampling complexity of topic models." This paper yields over an order of magnitude speedup in sampling topic models, in particular when the amount of data is large or when the generative model is dense.

June 1, 2014

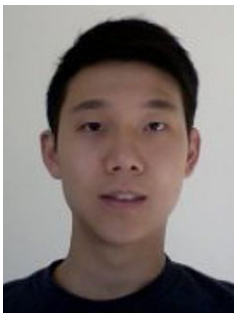
Siewiorek Recieves 2 DAC Awards



At the 51st ACM/IEEE Design and Automation Conference for electronic systems in San Francisco this year, Dan Siewiorek received two awards. The first was the Prolific Author Award for having published between 20 and 24 papers at the conference, and the second was the DAC's Second Decade (1974-1983) Award for its Top Ten Authors.

May 12, 2014

ISTC-CC Student Awarded Intel Foundation/SRCEA Graduate Fellowship



ECE doctoral student Kevin Kai-Wei Chang (CMU), who is working with Professor Onur Mutlu on efficient memory systems, has been selected to receive the prestigious Intel Foundation/SRCEA Graduate Fellowship. The fellowship provides tuition and a stipend for up to three years. Kevin recently published a paper at the HPCA 2014 conference on reducing the performance penalty of DRAM refresh, a key limiter of scalability in DRAM memory systems.

-- ECE News

May 2014

Mor Harchol-Balter Recipient of Two Teaching Awards

Congratulations to Mor Harchol-Balter (CMU) who received two awards as a result of her teachings (to 400 freshmen) of class 21-127 Proof Concepts: (i) 2014 CMU Mudge House Dinner with the Deans Honorary Event for Influential Teachers, and (ii) 2014 Apple Pie with Alpha Chi Honorary Event for CMU Faculty with Impact on Students.

April 21, 2014

Mutlu Receives Microsoft Research Award

ECE Professor Onur Mutlu has been selected as one of 12 applicants to receive a 2014 Microsoft Research Award from the Software Engineering Innovation Foundation (SEIF). The $40,000 award was granted for Mutlu's project "Improving Datacenter Efficiency and Total Cost of Ownership with Differentiated Software Reliability Analysis and Techniques."

--Inside CIT

April 15, 2014

Best Paper Award!

Onur Mutlu (CMU) and co-authors Hyoseung Kim, Dionisio de Niz, Bjorn Andersson, Mark Klein, and Ragunathan (Raj) Rajkumar received the Best Paper Award at the 20th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Berlin, Germany for their work on "Bounding Memory Interference Delay in COTS-based Multi-Core Systems."

April 9, 2014

ISTC-CC Paper Presented in Best Paper Session

Samira Khan, an ECE post-doctoral researcher, presented a paper at the 2014 International Symposium on High-Performance Computer Architecture (HPCA), during the Best Paper Session. Dr. Khan was lead author of "Improving Cache Performance by Exploiting Read-Write Disparity." The paper introduces a new mechanism that takes into account the differences in the performance cost of read and write operations in processor caches. It shows that designing a cache that prioritizes cache blocks that serve the more critical read operations can significantly improve system performance.

--ECE News

March 10, 2014

Best Paper Award!

Congratulations to Wolfgang Richter (CMU), Canturk Isci (IBM Research), Jan Harkes and Benjamin Gilbert (CMU), Vasanth Bala (IBM Research), and Mahadev Satyanarayan (CMU), who have won the International Conference on Cloud Engineering (IC2E) Best Paper Award for their paper entitled "Agentless Cloud-wide Streaming of Guest File System Updates."

March 4, 2014

Greg Ganger Receives 2014 Steven J. Fenves Award

ICES (CMU's The Institute for Complex Engineered Systems) has announced that Greg Ganger is the recipient of the 2014 Steven J. Fenves Award for Systems Research. He is the Jatras Professor of Electrical and Computer Engineering. He is also the director of the Parallel Data Lab.

ISTC-CC News

The "Steven J. Fenves Award for Systems Research" is presented annually to individuals for their contributions to systems research in areas that are relevant to the College of Engineering and ICES. The awards were formally presented at the CIT Faculty Awards Reception this spring.

Ganger is being recognized for his significant contributions to computer systems, in particular for his work on soft updates and self-* storage systems.

--ICES@CMU News

February 2014

ISTC-CC Students Receive Bertucci Fellowships

Congratulations to ISTC students Justin Meza and Lavanya Subramanian, who were awarded a John and Claire Bertucci Fellowship. The Bertucci Fellowships are awarded to accomplished graduate students who are pursuing doctoral degrees, have passed their PhD qualifying exams and have been admitted to PhD candidacy. The fellowships provide financial support towards their studies and research.

February 26, 2014

ISTC-CC Students Receive Award for Best Demo

"QuiltView: Glass-Sourced Video for Google Maps Queries," a demo presented by Zhuo Chen, Wenlu Hu, Kiryong Ha, Jan Harkes, Benjamin Gilbert, Jason Hong, Asim Smailagic, Dan Siewiorek, and Mahadev Satyanarayanan, received the Best Demo Award at the 15th International Workshop on Mobile Computing Systems and Applications (HotMobile'14).

December 2013

Gandhi Awarded SPEC Dissertation Award

Anshul Gandhi (ISTC-CC alum) won the SPEC Dissertation award for his Ph.D. thesis, titled, "Dynamic Server Provisioning for Data Center Power Management," awarded at the International conference on Performance Engineering in March 2014. The Research Group of the Standard Performance Evaluation Corporation (SPEC) selects the annual research prize to be awarded to a Ph.D. student whose thesis is regarded to be an exceptional, innovative contribution in the scope of the SPEC Research Group. Anshul is starting his new position this September as an Assistant Professor at SUNY StonyBrook.

November 25, 2013

3 ISTC-CC Members Elected IEEE Fellows

We are pleased to announce that 3 ISTC-CC faculty members have been selected as IEEE Fellows. Phillip Gibbons (Intel Labs) was selected for contributions to parallel computing and databases; Garth Gibson (CMU) was selected for contributions to the performance and reliability of transformative storage systems; and Sudhakar Yalamanchili (Georgia Institute of Technology) was selected for contributions to high-performance multiprocessor architecture and communication. Becoming an IEEE Fellow is a distinction reserved for select IEEE members whose extraordinary accomplishments in any of the IEEE fields of interest are deemed fitting of this prestigious grade elevation.

October 25, 2013

Joseph Gonzales Nominated for ACM Outstanding Dissertation

Congratulations to Joseph Gonzales (Advisor, Carlos Guestrin) who has been awarded the SCS Dissertation Award and nominated for ACM Outstanding Dissertation for his work on "Parallel and Distributed Systems for Probabilistic Reasoning."

September 18, 2013

New Thinking Track at SDC'13

In an effort to further cross-pollinate industry and academic research efforts, the Storage Developer Conference expanded its program to include a "New Thinking" track. A program committee of leading researchers compiled a list of 27 recent leading papers. The SNIA Technical Council then selected five of them to form the track at the conference.

Two ISTC-CC papers were presented: "LazyBase: Trading Freshness for Performance in a Scalable Database" by Cipar, Ganger, Keeton, Morrey, Soules, and Veitch, originally published at Eurosys '12, discusses scalable database systems specialized for the class of data analysis applications that extract knowledge from large, rapidly changing data sets.

"GraphChi: Large-Scale Graph Computation on Just a PC" by Kyrola, Blelloch and Guestrin, originally published at OSDI '12, proposes Parallel Sliding Windows, a novel method for efficiently processing large graphs from external memory (disk).

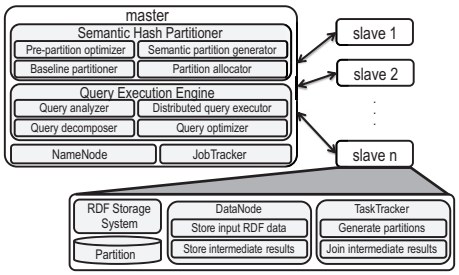
Recent Publications

Scaling Queries over Big RDF Graphs with Semantic Hash Partitioning

Kisung Lee, Ling Liu

To appear in Proceedings of the 40th IEEE International Conference on Very Large Databases (VLDB'14), Sept. 2014.

Massive volumes of big RDF data are growing beyond the performance capacity of conventional RDF data management systems operating on a single node. Applications using large RDF data demand efficient data partitioning solutions for supporting RDF data access on a cluster of compute nodes. In this paper we present a novel semantic hash partitioning approach and implement a Semantic HAsh Partitioning-Enabled distributed RDF data management system, called SHAPE. This paper makes three original contributions. First, the semantic hash partitioning approach we propose extends the simple hash partitioning method through direction-based triple groups and direction-based triple replications. The latter enhances the former by controlled data replication through intelligent utilization of data access locality, such that queries over big RDF graphs can be processed with zero or very small amount of inter-machine communication cost. Second, we generate locality-optimized query execution plans that are more efficient than popular multi-node RDF data management systems by effectively minimizing the inter-machine communication cost for query processing. Third but not the least, we provide a suite of locality-aware optimization techniques to further reduce the partition size and cut down on the inter-machine communication cost during distributed query processing. Experimental results show that our system scales well and can process big RDF datasets more efficiently than existing approaches.



System Architecture

Using RDMA Efficiently for Key-Value Services

Anuj Kalia, Michael Kaminsky, David G. Andersen

Proceedings of ACM SIGCOMM '14, August 2014.

This paper describes the design and implementation of HERD, a key-value system designed to make the best use of an RDMA network. Unlike prior RDMA-based key-value systems, HERD focuses its design on reducing network round trips while using efficient RDMA primitives; the result is substantially lower latency, and throughput that saturates modern, commodity RDMA hardware. HERD has two unconventional decisions: First, it does not use RDMA reads, despite the allure of operations that bypass the remote CPU entirely. Second, it uses a mix of RDMA and messaging verbs, despite the conventional wisdom that the messaging primitives are slow. A HERD client writes its request into the server's memory; the server computes the reply. This design uses a single round trip for all requests and supports up to 26 million key-value operations per second with 5 μs average latency. Notably, for small key-value items, our full system throughput is similar to native RDMA read throughput and is over 2X higher than recent RDMA-based key-value systems. We believe that HERD further serves as an effective template for the construction of RDMA-based datacenter services.

Efficient Coflow Scheduling with Varys

Mosharaf Chowdhury, Yuan Zhong, Ion Stoica

Proceedings of ACM SIGCOMM'14, August 2014.

Communication in data-parallel applications often involves a collection of parallel flows. Traditional techniques to optimize flowlevel metrics do not perform well in optimizing such collections, because the network is largely agnostic to application-level requirements. The recently proposed coflow abstraction bridges this gap and creates new opportunities for network scheduling. In this paper, we address inter-coflow scheduling for two different objectives: decreasing communication time of data-intensive jobs and guaranteeing predictable communication time. We introduce the concurrent open shop scheduling with coupled resources problem, analyze its complexity, and propose effective heuristics to optimize either objective. We present Varys, a system that enables data-intensive frameworks to use coflows and the proposed algorithms while maintaining high network utilization and guaranteeing starvation freedom. EC2 deployments and trace-driven simulations show that communication stages complete up to 3.16× faster on average and up to 2× more coflows meet their deadlines using Varys in comparison to per-flow mechanisms. Moreover, Varys outperforms non-preemptive coflow schedulers by more than 5×.

Phase-Concurrent Hash Tables for Determinism

Julian Shun, Guy Blelloch

Proceedings of the 26th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'14), June 2014.

We present a deterministic phase-concurrent hash table in which operations of the same type are allowed to proceed concurrently, but operations of different types are not. Phase-concurrency guarantees that all concurrent operations commute, giving a deterministic hash table state, guaranteeing that the state of the table at any quiescent point is independent of the ordering of operations. Furthermore, by restricting our hash table to be phase-concurrent, we show that we can support operations more efficiently than previous concurrent hash tables. Our hash table is based on linear probing, and relies on history-independence for determinism.

We experimentally compare our hash table on a modern 40-core machine to the best existing concurrent hash tables

Recent Publications

that we are aware of (hopscotch hashing and chained hashing) and show that we are 1.3–4.1 times faster on random integer keys when operations are restricted to be phase-concurrent. We also show that the cost of insertions and deletions for our deterministic hash table is only slightly more expensive than for a non-deterministic version that we implemented. Compared to standard sequential linear probing, we get up to 52 times speedup on 40 cores with dual hyperthreading. Furthermore, on 40 cores insertions are only about 1:3 slower than random writes (scatter). We describe several applications which have deterministic solutions using our phase-concurrent hash table, and present experiments showing that using our phase-concurrent deterministic hash table is only slightly slower than using our non-deterministic one and faster than using previous concurrent hash tables, so the cost of determinism is small.

Will They Blend?: Exploring Big Data Computation atop Traditional HPC NAS Storage

Ellis Wilson, Mahmut Kandemir, Garth Gibson

Proceedings of 34th IEEE International Conference on Distributed Computing Systems (ICDCS'14), June-July 2014.

The Apache Hadoop framework has rung in a new era in how data-rich organizations can process, store, and analyze large amounts of data. This has resulted in increased potential for an infrastructure exodus from the traditional solution of commercial database ad-hoc analytics on network-attached storage (NAS). While many data-rich organizations can afford to either move entirely to Hadoop for their Big Data analytics, or to maintain their existing traditional infrastructures and acquire a new set of infrastructure solely for Hadoop jobs, most supercomputing centers do not enjoy either of those possibilities. Too much of the existing scientific code is tailored to work on massively parallel file systems unlike the Hadoop Distributed File System (HDFS), and their datasets are too large to reasonably maintain and/or ferry between two distinct storage systems. Nevertheless, as scientists search for easier-to-program frameworks with a lower time-to-science to post-process their huge datasets after execution, there is increasing pressure to enable use of MapReduce within these traditional High Performance Computing (HPC) architectures.

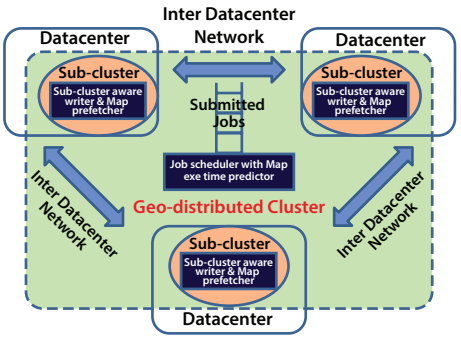
Therefore, in this work we explore potential means to enable use of the easy-to-program Hadoop MapReduce framework without requiring a complete infrastructure overhaul from existing HPC NAS solutions. We demonstrate that retaining function-dedicated resources like NAS is not only possible, but can even be effected efficiently with MapReduce. In our exploration, we unearth subtle pitfalls resultant from this mashup of new-era Big Data computation on conventional HPC storage and share the clever architectural configurations that allow us to avoid them. Last, we design and present a novel Hadoop File System, the Reliable Array of Independent NAS File System (RainFS), and experimentally demonstrate its improvements in performance and reliability over the previous architectures we have investigated.

Improving Hadoop Service Provisioning in A Geographically Distributed Cloud

Qi Zhang, Ling Liu, Aameek Singh, Nagapramod Mandagere, Sandeep Gopisetty, Gabriel Alatorre, Kisung Lee, Yang Zhou

Proceedings of IEEE 7th Int'l. Conference on Cloud Computing (Cloud'14), June-July 2014.

With more data generated and collected in a geographically distributed manner, combined by the increased computational requirements for large scale data-intensive analysis, we have witnessed the growing demand for geographically distributed Cloud datacenters and hybrid Cloud service provisioning, enabling organizations to support instantaneous demand of additional computational resources and to expand inhouse resources to maintain peak service demands by utilizing cloud resources. A key challenge for



Architecture of a geo-distributed cluster

running applications in such a geographically distributed computing environment is how to efficiently schedule and perform analysis over data that is geographically distributed across multiple datacenters. In this paper, we first compare multi-datacenter Hadoop deployment with single-datacenter Hadoop deployment to identify the performance issues inherent in a geographically distributed cloud. A generalization of the problem characterization in the context of geographically distributed cloud datacenters is also provided with discussions on general optimization strategies. Then we describe the design and implementation of a suite of system-level optimizations for improving performance of Hadoop service provisioning in a geo-distributed cloud, including prediction-based job localization, configurable HDFS data placement, and data prefetching. Our experimental evaluation shows that our prediction based localization has very low error ratio, smaller than 5%, and our optimization can improve the execution time of Reduce phase by 48.6%.

GraphLens: Mining Enterprise Storage Workloads Using Graph Analytics

Yang Zhou, Sangeetha Seshadri, Larry Chiu, Ling Liu

IEEE 2nd International Congress on Big Data (Big Data'14), June-July 2014.

Conventional methods used to analyze storage workloads have been centered on relational database technology combined with attributes-based classification algorithms. This paper

continued on pg. 10

Recent Publications

continued from pg. 9

presents a novel analytic architecture, GraphLens, for mining and analyzing real world storage traces. The design of our GraphLens system embodies three unique features.

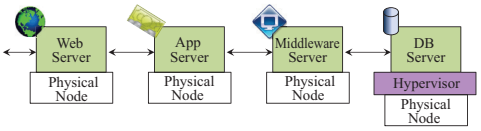
First, we model storage traces as heterogeneous trace graphs in order to capture diverse spatial correlations and storage access patterns using a unified analytic framework. Second, we employ and develop an innovative graph clustering method to discover interesting spatial access patterns. This enables us to better characterize important hotspots of storage access and understand hotspot movement patterns. Third, we design a unified weighted similarity measure through an iterative learning and dynamic weight refinement algorithm. With an optimal weight assignment scheme, we can efficiently combine the correlation information for each type of storage access patterns, such as random vs. sequential, read vs. write, to identify interesting spatial correlations hidden in the traces. Extensive evaluation on real storage traces shows GraphLens can provide scalable and reliable data analytics for better storage strategy planning and efficient data placement guidance.

IO Performance Interference among Consolidated n-Tier Applications Sharing is Better than Isolation, Again

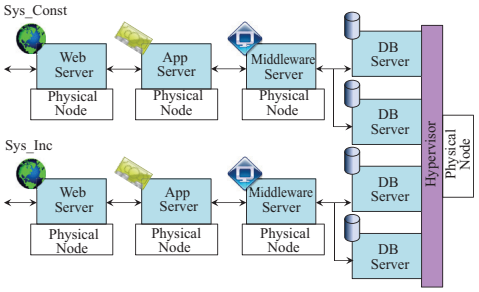
Chien-An Lai, Qingyang Wang, Josh Kimball, Jack Li, Junhee Park, Calton Pu

Proceedings of IEEE 7th Int. Conf. on Cloud Computing (Cloud'14), June-July 2014.

The performance unpredictability associated with migrating applications into cloud computing infrastructures has impeded this migration. For example, CPU contention between co-located applications has been shown to exhibit counter-intuitive behavior. In this paper, we investigate IO performance interference through the experimental study of consolidated n-tier applications leveraging the same disk. Surprisingly, we found that specifying a specific disk allocation, e.g., limiting the number of Input/ Output Operations Per Second (IOPs) per VM, results in significantly lower performance than fully sharing disk across VMs. Moreover, we observe severe performance interference among VMs can not be totally eliminated even with a sharing strategy (e.g., response times for constant workloads still increase over 1,100%). By using a micro-benchmark (Filebench) and an n-tier application benchmark systems (RUBBoS), we demonstrate the existence of disk contention in consolidated environments, and how performance loss occurs when co-located database systems in order to maintain database consistency flush their logs from memory to disk. Potential solutions to these isolation issues are (1) to increase the log buffer size to amortize the disk IO cost (2) to decrease the number of write threads to alleviate disk contention. We validate these methods experimentally and find a 64% and 57% reduction in response time (or more generally, a reduction in performance interference) for constant and increasing workloads respectively.



(a) Dedicated deployment of a 4-tier application system with four software servers (i.e., web, application, middleware, and database) and four physical hardware nodes



(b) Consolidated deployment of two 4-tier systems (Sys Const and Sys Inc) with 1/1/1/2 configuration and seven physical hardware nodes in total. The DB server are co-ocated in dedicated VMs on a single shared physical hardware node.

Example of a dedicated 1(a) and a consolidated 1(b) 4-tier application system deployment, presented as mappings of software servers to physical hardware nodes.

Improving MapReduce Performance in a Heterogeneous Cloud: A Measurement Study

Xu Zhao, Ling Liu, Qi Zhang, Xiaoshe Dong

Proceedings of IEEE 7th Int. Conf. on Cloud Computing (Cloud'14), June-July 2014.

Hybrid clouds, geo-distributed cloud and continuous upgrades of computing, storage and networking resources in the cloud have driven datacenters evolving towards heterogeneous clusters. Unfortunately, most of MapReduce implementations are designed for homogeneous computing environments and perform poorly in heterogeneous clusters. Although a fair of research efforts have dedicated to improve MapReduce performance, there still lacks of in-depth understanding of the key factors that affect the performance of MapReduce jobs in heterogeneous clusters. In this paper, we present an extensive experimental study on two categories of factors: system configuration and task scheduling. Our measurement study shows that an in-depth understanding of these factors is critical for improving MapReduce performance in a heterogeneous environment. We conclude with five key findings: (1) Early shuffle, though effective for reducing the latency of MapReduce jobs, can impact the performance of map tasks and reduce tasks differently when running on different types of nodes. (2) Two phases in map tasks have different sensitive to input block size and the ratio of sort phase with different block size is different for different type of nodes. (3) Scheduling map or reduce tasks dynamically with node capacity and workload awareness can further enhance the job performance and improve resource consumption efficiency. (4) Although random scheduling of reduce tasks works well in homogeneous clusters, it can significantly degrade the performance in heterogeneous clusters when shuffled data size is large. (5) Phase-aware progress rate estimation and speculation strategy can provide substantial performance gain over the state of art speculation scheduler.

The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study

Samira Khan, Donghyuk Lee, Yoongu Kim, Alaa Alameldeen, Chris Wilkerson, Onur Mutlu

Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'14), June 2014.

As DRAM cells continue to shrink, they become more susceptible to retention failures. DRAM cells that permanently exhibit short retention times are fairly easy to identify and repair through the use of memory tests and row and column redundancy. However, the retention time of many cells may vary over time due to a property called Variable Retention Time (VRT). Since these cells intermittently transition between failing and non-failing states, they are particularly difficult to identify through memory tests alone. In addition, the high temperature packaging process may aggravate this problem as the susceptibility of cells to VRT increases after the assembly of DRAM chips. A promising alternative to manufacture-time testing is to detect and mitigate retention failures after the system has become operational. Such a system would require mechanisms to detect and mitigate retention failures in the field, but would be responsive to retention failures introduced after system assembly and could dramatically reduce the cost of testing, enabling much longer tests than are practical with manufacturer testing equipment.

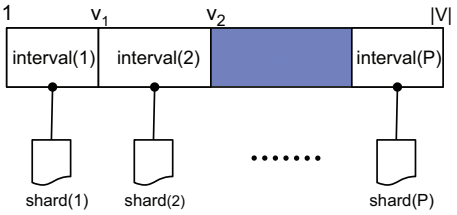
In this paper, we analyze the efficacy of three common error mitigation techniques (memory tests, guardbands, and error correcting codes (ECC)) in real DRAM chips exhibiting both intermittent and permanent retention failures. Our analysis allows us to quantify the efficacy of recent system-level error mitigation mechanisms that build upon these techniques. We revisit prior works in the context of the experimental data we present, showing that our measured results significantly impact these works' conclusions. We find that mitigation techniques that rely on run-time testing alone [38, 27, 50, 26] are unable to ensure reliable operation even after many months of testing. Techniques that incorporate ECC [4, 52], however, can ensure reliable DRAM operation after only a few hours of testing. For example, VS-ECC [4], which couples testing with variable strength codes to allocate the strongest codes to the most error-prone memory regions, can ensure reliable operation for 10 years after only 19 minutes of testing. We conclude that the viability of these mitigation techniques depend on efficient online profiling of DRAM performed without disrupting system operation.

Beyond Synchronous: New Techniques for External Memory Graph Algorithms

Aapo Kyrola, Julian Shun, Guy Blelloch

Proceedings of the Symposium on Experimental Algorithms (SEA'14), June 2014.

GraphChi [16] is a recent high-performance system for external memory (disk-based) graph computations. It uses the Parallel Sliding Windows (PSW) algorithm which is based on the so-called Gauss-Seidel type of iterative computation, in which updates to values are immediately visible within the iteration. In contrast, previous external memory graph algorithms are based on the synchronous model where computation can only observe values from previous iterations. In this work, we study implementations of connected components and minimum spanning forest on PSW and show that they have a competitive I/O bound of O(sort(E) log(V/M)) and also work well in practice. We also show that our MSF implementation is competitive with a specialized algorithm proposed by Dementiev et al. [10] while being much simpler.




```
1: procedure PSW (G, updateFunc)
2:     for interval I_i ⊂ V do
3:         G_i := LoadSubgraph(I_i)
4:         for v ∈ G_i.V do
5:             updateFunc(v, G_i.E[v])
6:         UpdateToDisk(G_i)
```

Top: The vertices of graph (V;E) are divided into P intervals. Each interval is associated with a shard, which stores all edges that have destination vertex in that interval. Bottom: Pseudo-code for the main loop of Parallel Sliding Windows. Note that both for-loops can iterate in random order.

Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories

Yu Cai, Gulay Yalcin, Onur Mutlu, Eric Haratsch, Osman Unsal, Adrian Cristal, Ken Mai

Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'14), June 2014.

Continued scaling of NAND flash memory to smaller process technology nodes decreases its reliability, necessitating more sophisticated mechanisms to correctly read stored data values. To distinguish between different potential stored values, conventional techniques to read data from flash memory employ a single set of reference voltage values, which are determined based on the overall threshold voltage distribution of flash cells. Unfortunately, the phenomenon of program interference, in which a cell's threshold voltage unintentionally changes when a neighboring cell is programmed, makes this conventional approach increasingly inaccurate in determining the values of cells.

This paper makes the new empirical observation that identifying the value stored in the immediate-neighbor cell makes it easier to determine the data value stored in the cell that is being read. We provide a detailed statistical and experimental characterization of threshold voltage distribution of flash memory cells conditional upon the immediate-neighbor cell values, and

continued on pg. 12

Recent Publications

continued from pg. 11

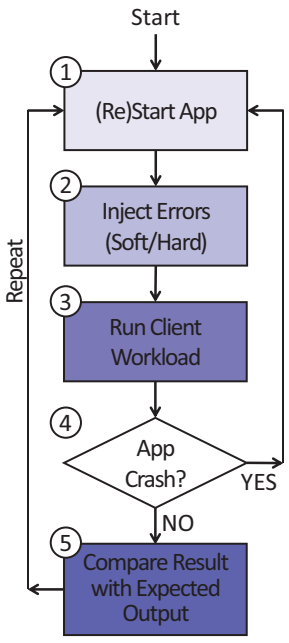
show that such conditional distributions can be used to determine a set of read reference voltages that lead to error rates much lower than when a single set of reference voltage values based on the overall distribution are used. Based on our analyses, we propose a new method for correcting errors in a flash memory page, neighborcell assisted correction (NAC). The key idea is to re-read a flash memory page that fails error correction codes (ECC) with the set of read reference voltage values corresponding to the conditional threshold voltage distribution assuming a neighbor cell value and use the re-read values to correct the cells that have neighbors with that value. Our simulations show that NAC effectively improves flash memory lifetime by 33% while having no (at nominal lifetime) or very modest (less than 5% at extended lifetime) performance overhead.

Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost via Heterogeneous-Reliability Memory

Yixin Luo, Sriram Govindan, Bikash Sharma, Mark Santaniello, Justin Meza, Aman Kansal, Jie Liu, Badriddine Khessib, Kushagra Vaid, Onur Mutlu

Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'14), June 2014.

Memory devices represent a key component of datacenter total cost of ownership (TCO), and techniques used to reduce errors that occur on these devices increase this cost. Existing approaches to providing reliability for memory devices pessimistically treat all data as equally vulnerable to memory errors. Our key insight is that there exists a diverse spectrum of tolerance to memory errors in new data-intensive applications, and that traditional one-size-fits-all memory reliability techniques are inefficient in terms of cost. For example, we found that while traditional error protection increases memory system cost by



Memory error emulation framework.

12.5%, some applications can achieve 99.00% availability on a single server with a large number of memory errors without any error protection. This presents an opportunity to greatly reduce server hardware cost by provisioning the right amount of memory reliability for different applications.

Toward this end, in this paper, we make three main contributions to enable highly-reliable servers at low datacenter cost. First, we develop a new methodology to quantify the tolerance of applications to memory errors. Second, using our methodology, we perform a case study of three new data-intensive workloads (an interactive web search application, an in-memory key–value store, and a graph mining framework) to identify new insights into the nature of application memory error vulnerability. Third, based on our insights, we propose several new hardware/software heterogeneous-reliability memory system designs to lower datacenter cost while achieving high reliability and discuss their trade-os. We show that our new techniques can reduce server hardware cost by 4.7% while achieving 99.90% single server availability.

Towards Wearable Cognitive Assistance

Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, Mahadev Satyanarayanan

Proceedings of the 12th ACM International Conference on Mobile Computing, Systems and Services (MobiSys'14), June 2014.

We describe the architecture and prototype implementation of an assistive system based on Google Glass devices for users in cognitive decline. It combines the first-person image capture and sensing capabilities of Glass with remote processing to perform real-time scene interpretation. The system architecture is multi-tiered. It offers tight end-to-end latency bounds on compute-intensive operations, while ad- dressing concerns such as limited battery capacity and limited processing capability of wearable devices. The system gracefully degrades services in the face of network failures and unavailability of distant architectural tiers.

Knowing When You're Wrong: Building Fast and Reliable Approximate Query Processing Systems

Sameer Agarwal, Henry Milner, Ariel Kleiner, Ameet Talwalkar, Barzan Mozafari, Michael Jordan, Samuel Madden, Ion Stoica

Proceedings of ACM SIGMOD (SIGMOD'14), June 2014.

Modern data analytics applications typically process massive amounts of data on clusters of tens, hundreds, or thousands of machines to support near-real-time decisions. The quantity of data and limitations of disk and memory bandwidth often make it infeasible to deliver answers at interactive speeds. However, it has been widely observed that many applications can tolerate some degree of inaccuracy. This is especially true for exploratory queries on data, where users are satisfied with "close-enough" answers if they can come quickly. A popular technique for speeding up queries at the cost of accuracy is to execute each query on a

sample of data, rather than the whole dataset. To ensure that the returned result is not too inaccurate, past work on approximate query processing has used statistical techniques to estimate "error bars" on returned results. However, existing work in the sampling-based approximate query processing (S-AQP) community has not validated whether these techniques actually generate accurate error bars for real query workloads. In fact, we find that error bar estimation often fails on real world production workloads. Fortunately, it is possible to quickly and accurately diagnose the failure of error estimation for a query. In this paper, we show that it is possible to implement a query approximation pipeline that produces approximate answers and reliable error bars at interactive speeds.

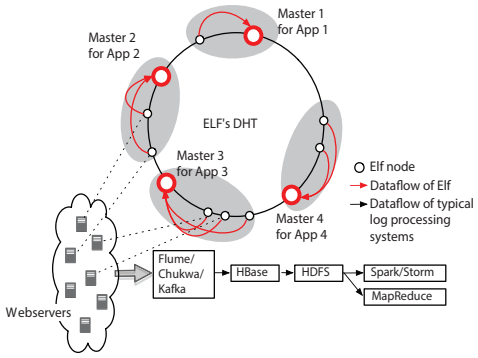
ELF: Efficient Lightweight Fast Stream Processing at Scale

Liting Hu, Karsten Schwan, Hrishikesh Amur, Xin Chen

20th USENIX Annual Technical Conference (ATC'14), June 2014.

Stream processing has become a key means for gaining rapid insights from webserver-captured data. Challenges include how to scale to numerous, concurrently running streaming jobs, to coordinate across those jobs to share insights, to make online changes to job functions to adapt to new requirements or data characteristics, and for each job, to efficiently operate over different time windows.

The ELF stream processing system addresses these new challenges. Implemented over a set of agents enriching the web tier of datacenter systems, ELF obtains scalability by using a decentralized "many masters" architecture where for each job, live data is extracted directly from webservers, and placed into memory-efficient compressed buffer trees (CBTs) for local parsing and temporary storage, followed by subsequent aggregation using shared reducer trees (SRTs) mapped to sets of worker processes. Job masters at the roots of SRTs can dynamically customize worker actions, obtain aggregated results for end user



Dataflow of ELF vs. a typical realtime web log analysis system, composed of Flume, HBase, HDFS, Hadoop MapReduce and Spark/Storm.

delivery and/or coordinate with other jobs. An ELF prototype implemented and evaluated for a larger scale configuration demonstrates scalability, high per-node throughput, sub-second job latency, and subsecond ability to adjust the actions of jobs being run.

Gleaner: Mitigating the Blocked-Waiter Wakeup Problem for Virtualized Multicore Applications

Xiaoning Ding, Phillip B. Gibbons, Michael A. Kozuch, Jianchen Shan

20th USENIX Annual Technical Conference (ATC'14), June 2014.

As the number of cores in a multicore node increases in accordance with Moore's law, the question arises as to what are the costs of virtualized environments when scaling applications to take advantage of larger core counts. While a widely-known cost due to preempted spinlock holders has been extensively studied, this paper studies another cost, which has received little attention. The cost is caused by the intervention from the VMM during synchronization-induced idling in the application, guest OS, or supporting libraries—we call this the blocked-waiter wakeup (BWW) problem.

The paper systematically analyzes the cause of the BWW problem and studies its performance issues, including increased execution times, reduced system throughput, and performance unpredictability. To deal with these issues, the paper proposes a solution, Gleaner, which integrates idling operations and imbalanced scheduling as a mitigation to this problem. We show how Gleaner can be implemented without intrusive modification to the guest OS. Extensive experiments show that Gleaner can effectively reduce the virtualization cost incurred by blocking synchronization and improve the performance of individual applications by 16x and system throughput by 3x.

Toward Combining Online & Offline Management of Big Data Applications

Brian Laub, Chengwei Wang, Karsten Schwan, Chad Huneycutt

MBDS Track, ACM International Conference on Autonomic Computing (ICAC'14), June 2014.

Traditional data center monitoring systems focus on collecting basic metrics such as CPU and memory usage, in a centralized location, giving administrators a summary of global system health via a database of observations. Conversely, emerging research systems are focusing on scalable, distributed monitoring capable of quickly detecting and alerting administrators to anomalies. This paper outlines VStore, a system that seeks to combine fast online anomaly detection with offline storage and analysis of monitoring data. VStore can be used as a historical reference to help guide administrators towards quickly classifying and fixing anomalous behavior once a problem has been detected. We demonstrate this idea with a distributed big streaming data application, and explore three common fault scenarios in this application. We show that each scenario exhibits a slightly different monitoring history, which may be undetectable by online algorithms that are resource-constrained. We also offer a discussion of how historical data captured by VStore can be combined with online monitoring tools to improve troubleshooting efforts in the data center.

continued on pg. 14

Recent Publications

continued from pg. 13

Don't Settle for Eventual Consistency

Wyatt Lloyd, Michael J. Freedman, Michael Kaminsky, David G. Andersen

Communications of the ACM (CACM), 57(5), May 2014.

Geo-replicated, distributed data stores that support complex online applications, such as social networks, must provide an "always-on" experience where operations always complete with low latency. Today's systems often sacrifice strong consistency to achieve these goals, exposing inconsistencies to their clients and necessitating complex application logic. In this paper, we identify and define a consistency model—causal consistency with convergent conflict handling, or causal+—that is the strongest achieved under these constraints.

We present the design and implementation of COPS, a key-value store that delivers this consistency model across the wide-area. A key contribution of COPS is its scalability, which can enforce causal dependencies between keys stored across an entire cluster, rather than a single server like previous systems. The central approach in COPS is tracking and explicitly checking whether causal dependencies between keys are satisfied in the local cluster before exposing writes. Further, in COPS-GT, we introduce get transactions in order to obtain a consistent view of multiple keys without locking or blocking. Our evaluation shows that COPS completes operations in less than a millisecond, provides throughput similar to previous systems when using one server per cluster, and scales well as we increase the number of servers in each cluster. It also shows that COPS-GT provides similar latency, throughput, and scaling to COPS for common workloads.

The Dirty-Block Index

Vivek Seshadri, Abhishek Bhowmick Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, Todd C. Mowry

Proceedings of the 41st International Symposium on Computer Architecture (ISCA'14), June 2014.

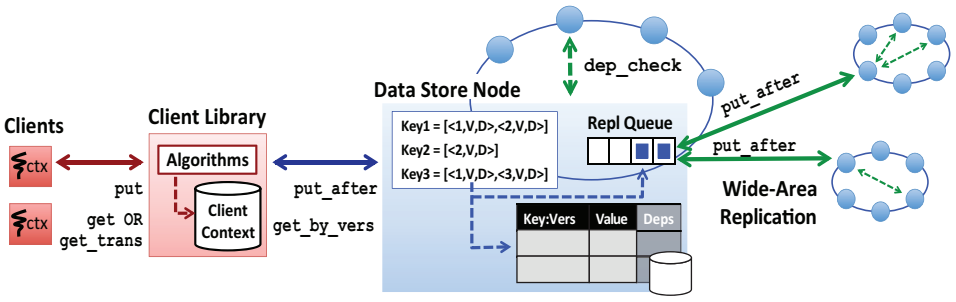
On-chip caches maintain multiple pieces of metadata about each cached block—e.g., dirty bit, coherence information, ECC. Traditionally, such metadata for each block is stored in the corresponding tag entry in the tag store. While this approach is simple to implement and scalable, it necessitates a full tag store lookup for any metadata query—resulting in high latency and energy consumption. We find that this approach is inefficient and inhibits several cache optimizations. In this work, we propose a new way of organizing the dirty bit information that enables simpler and more efficient implementations of several optimizations. In our proposed approach, we remove the dirty bits from the tag store and organize it differently in a separate structure, which we call the Dirty-Block Index (DBI). The organization of DBI is simple: it consists of multiple entries, each corresponding to some row in DRAM. A bit vector in each entry tracks whether or not each block in the corresponding DRAM row is dirty. We demonstrate the benefits of DBI by using it to simultaneously and efficiently implement three optimizations proposed by prior work: 1) Aggressive DRAM-aware writeback, 2) Bypassing cache lookups, and 3) Heterogeneous ECC for clean/dirty blocks. DBI, with all three optimizations enabled, improves performance by 31% compared to the baseline (by 6% compared to the best previous mechanism) while reducing overall cache area cost by 8% compared to prior approaches.

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, Onur Mutlu

Proceedings of the 41st International Symposium on Computer Architecture (ISCA'14), June 2014.

Memory isolation is a key property of a reliable and secure computing system—an access to one memory address should not have unintended side effects on data stored in other addresses. However, as DRAM process technology scales down to smaller dimensions, it becomes more difficult to prevent DRAM cells from electrically interacting with each other. In this paper, we expose the vulnerability of commodity DRAM chips to disturbance errors. By reading from the same address in DRAM, we show that it is possible to corrupt data in nearby addresses. More specifically, activating the same row in DRAM corrupts data in nearby rows. We demonstrate this phenomenon on Intel and AMD systems using a malicious program that generates many DRAM accesses. We induce errors in most DRAM modules (110 out of 129) from three major DRAM manufacturers. From this we conclude that many deployed systems are likely to be at risk. We identify the root cause of disturbance errors as the repeated toggling of a DRAM row's wordline, which stresses inter-cell coupling effects that accelerate charge leakage from nearby rows. We provide



The COPS architecture. A client library exposes a put/get interface to its clients and ensures operations are properly labeled with causal dependencies. A key-value store replicates data between clusters, ensures writes are committed in their local cluster only after their dependencies have been satisfied, and in COPS-GT, stores multiple versions of each key along with dependency metadata.

an extensive characterization study of disturbance errors and their behavior using an FPGA-based testing platform. Among our key findings, we show that (i) it takes as few as 139K accesses to induce an error and (ii) up to one in every 1.7K cells is susceptible to errors. After examining various potential ways of addressing the problem, we propose a low-overhead solution to prevent the errors.

Exact Analysis of the M/M/k/setup Class of Markov Chains via Recursive Renewal Reward

Anshul Gandhi, Sherwin Doroudi, Mor Harchol-Balter, Alan Scheller-Wolf

Queueing Systems: Theory and Applications Vol. 77, No. 2, 2014, pp. 177-209. June 2014.

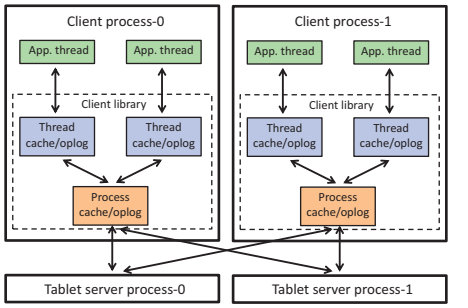
The M/M/k/setup model, where there is a penalty for turning servers on, is common in data centers, call centers, and manufacturing systems. Setup costs take the form of a time delay, and sometimes there is additionally a power penalty, as in the case of data centers. While the M/M/1/setup was exactly analyzed in 1964, no exact analysis exists to date for the M/M/k/setup with k>1. In this paper, we provide the first exact, closed-form analysis for the M/M/k/setup and some of its important variants including systems in which idle servers delay for a period of time before turning off or can be put to sleep. Our analysis is made possible by a new way of combining renewal reward theory and recursive techniques to solve Markov chains with a repeating structure. Our renewal-based approach uses ideas from renewal reward theory and busy period analysis to obtain closed-form expressions for metrics of interest such as the transform of time in system and the transform of power consumed by the system. The simplicity, intuitiveness, and versatility of our renewal-based approach makes it useful for analyzing Markov chains far beyond the M/M/k/setup. In general, our renewal-based approach should be used to reduce the analysis of any 2-dimensional Markov chain which is infinite in at most one dimension and repeating to the problem of solving a system of polynomial equations. In the case where all transitions in the repeating portion of the Markov chain are skip-free and all up/down arrows are unidirectional, the resulting system of equations will yield a closed-form solution.

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

20th USENIX Annual Technical Conference (ATC'14), June 2014.

Many modern machine learning (ML) algorithms are iterative, converging on a final solution via many iterations over the input data. This paper explores approaches to exploiting these algorithms' convergent nature to improve performance, by allowing parallel and distributed threads to use loose consistency models for shared algorithm state. Specifically, we focus on bounded staleness, in which each thread can see a view of the current intermediate solution that may be a limited number of iterations out-of-date. Allowing staleness reduces communication costs (batched updates and cached reads) and synchronization (less waiting for locks or straggling threads). One approach is to increase the number of iterations between barriers in the oft-used Bulk Synchronous Parallel (BSP) model of parallelizing, which mitigates these costs when all threads proceed at the same speed. A more flexible approach, called Stale Synchronous Parallel (SSP), avoids barriers and allows threads to be a bounded number of iterations ahead of the current slowest thread. Extensive experiments with ML algorithms for topic modeling, collaborative filtering, and PageRank show that both approaches significantly increase convergence speeds, behaving similarly when there are no stragglers, but SSP outperforms BSP in the presence of stragglers.



LazyTable running two application processes with two application threads each.

The Cost of Fault Tolerance in Multi-Party Communication Complexity.

Binbin Chen, Haifeng Yu, Yuda Zhao, Phillip B. Gibbons

Journal of the ACM, May 2014.

Multi-party communication complexity involves distributed computation of a function over inputs held by multiple distributed players. A key focus of distributed computing research, since the very beginning, has been to tolerate failures. It is thus natural to ask "If we want to compute a certain function in a fault-tolerant way, what will the communication complexity be?" For this question, this article will focus specifically on (i) tolerating node crash failures, and (ii) computing the function over general topologies (instead of, e.g., just cliques). One way to approach this question is to first develop results in a simpler failure-free setting, and then "amend" the results to take into account failures' impact.

Whether this approach is effective largely depends on how big a difference failures can make. This article proves that the impact of failures is significant, at least for the SUM aggregate function in general topologies: As our central contribution, we prove that there exists (at least) an exponential gap between the non-fault-tolerant and fault-tolerant communication complexity of SUM. This gap attests that fault-tolerant communication complexity needs to be studied separately from non-fault-tolerant communication complexity, instead of being considered as an "amended" version of the latter. Such exponential gap is

continued on pg. 16

Recent Publications

continued from pg. 15

not obvious: For some other functions such as the MAX aggregate function, the gap is only logarithmic.

Part of our results are obtained via a novel reduction from a new two-party problem UNIONSIZECP that we introduce. UNIONSIZECP comes with a novel cycle promise, which is the key enabler of our reduction. We further prove that this cycle promise and UNIONSIZECP likely play a fundamental role in reasoning about fault-tolerant communication complexity.

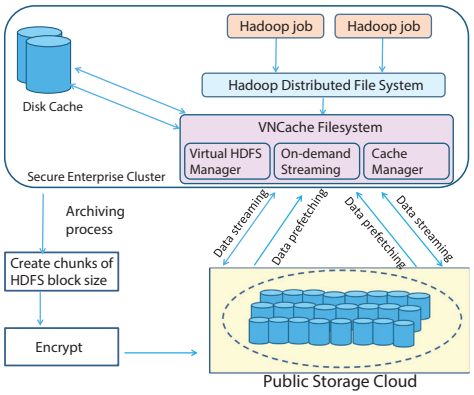
MapReduce Analysis for Cloud-archived Data

Balaji Palanisamy, Aameek Singh, Nagapramod Mandagere, Gabriel Alatorre, Ling Liu

Proceedings of the 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'14), May 2014.

Public storage clouds have become a popular choice for archiving certain classes of enterprise data - for example, application and infrastructure logs. These logs contain sensitive information like IP addresses or user logins due to which regulatory and security requirements often require data to be encrypted before moved to the cloud. In order to leverage such data for any business value, analytics systems (e.g. Hadoop/MapReduce) first download data from these public clouds, decrypt it and then process it at the secure enterprise site.

We propose VNCache: an efficient solution for MapReduce analysis of such cloud-archived log data without requiring an a priori data transfer and loading into the local Hadoop cluster. VNcache dynamically integrates cloud-archived data into a virtual namespace at the enterprise Hadoop cluster. Through a seamless data streaming and prefetching model, Hadoop jobs can begin execution as soon as they are launched without requiring any a priori downloading. With VNcache's accurate prefetching and caching, jobs often run on a local cached copy of the data block significantly improving performance. When no longer needed,



VNCache system model.

data is safely evicted from the enterprise cluster reducing the total storage footprint. Uniquely, VNcache is implemented with NO changes to the Hadoop application stack.

A Technology Probe of Wearable In-Home Computer-Assisted Physical Therapy

Kevin Huang, Patrick J. Sparto, Sara Kiesler, Asim Smailagic, Jennifer Mankoff, Dan Siewiorek

The 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI'14), May 2014.

Physical therapists could make better treatment decisions if they had accurate patient home exercise data but today this information is only available from patient self-report. A more accurate source of data could be gained from wearable computing designed for physical therapy exercise support. Existing systems have been tested in the lab but we have little information about issues they may face in home settings. We designed a technology probe, SenseCap, and deployed it for seven days in ten physical therapy patients' homes. SenseCap is a wearable physical therapy support system that gathers patient exercise compliance and performance data and summarizes the data in charts on an iPad Dashboard for physical therapists to view when patients return to the clinic. In this paper, we present the results of our deployment, show in-home patient exercise data gathered by the probe, and make design recommendations based on patient and physical therapist responses.

Exploring Graph Analytics for Cloud Troubleshooting

Chengwei Wang, Karsten Schwan, Brian Laub, Mukil Kesavan, Ada Gavrilovska

International Conference on Autonomic Computing (ICAC'14), ACM short paper, June 2014.

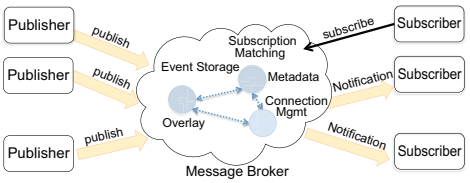
We propose VFocus, a platform which uses streaming graph analytics to narrow down the search space for troubleshooting and management in large scale data centers. This paper describes useful guidance operations which are realized with graph analytics and validated with representative use cases. The first case is based on real data center traces to measure the performance of troubleshooting operations supported by VFocus. In the second use case, the utility of VFocus is demonstrated by detecting data hotspots in a big data stream processing application. Experimental results show that VFocus guidance operations can troubleshoot Virtual Machine (VM) migration failures with accuracy of 83% and with delays of only hundreds of milliseconds when tracking migrations on 256 servers hosing 1024 VMs. Such successes are achieved with negligible runtime overheads and low perturbation for applications, in comparison to brute-force approaches.

Flexpath: Type-Based Publish/Subscribe System for Large-scale Science Analytics

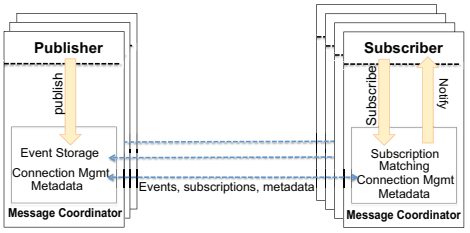
Jai Dayal, Drew Bratcher, Greg Eisenhauer, Karsten Schwan, Matthew Wolf, Xuechen Zhang, Hasan Abbasi, Scott Klasky, Norbert Podhorszki

Proceedings of the 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'14), May 2014.

As high-end systems move toward exascale sizes, a new model of scientific inquiry being developed is one in which online data analytics run concurrently with the high end simulations



(a) Traditional Pub/Sub



(b) Flexpath Pub/Sub

Traditional model of publish/subscribe vs. Flexpath model of publish/subscribe allowing for fine-grained data exchanges across parallel applications.

producing data outputs. Goals are to gain rapid insights into the ongoing scientific processes, assess their scientific validity, and/or initiate corrective or supplementary actions by launching additional computations when needed. The Flexpath system presented in this paper addresses the fundamental problem of how to structure and efficiently implement the communications between high end simulations and concurrently running online data analytics, the latter comprised of componentized dynamic services and service pipelines.

Using a type-based publish/subscribe approach, Flexpath encourages diversity by permitting analytics services to differ in their computational and scaling characteristics and even in their internal execution models. Flexpath uses direct and MxN connections between interacting services to reduce data movements, to allow for runtime connectivity changes to accommodate component arrivals/departures, and to support the multiple underlying communication protocols used for analytics workflows in which simulation outputs are processed by analytics services residing on the same nodes where they are generated, on the same machine, and/or on attached or remote analytics engines. This paper describes the design and implementation of Flexpath, and evaluates it with two widely used scientific applications and their associated data analytics methods.

Attack-resilient Mix-zones over Road Networks: Architecture and Algorithms

Balaji Palanisamy, Ling Liu

IEEE Transactions on Mobile Computing (TMC), May 2014.

Continuous exposure of location information, even with spatially cloaked resolution, may lead to breaches of location privacy due to statistics-based inference attacks. An alternative and complementary approach to spatial cloaking based location anonymization is to break the continuity of location exposure by introducing techniques, such as mix-zones, where no application can trace user movements. Several factors impact on the effectiveness of mix-zone approach, such as user population, mix-zone geometry, location sensing rate and spatial resolution, as well as spatial and temporal constraints on user movement patterns. However, most of the existing mix-zone proposals fail to provide effective mix-zone construction and placement algorithms that are resilient to timing and transition attacks. This paper presents MobiMix, a road network based mix-zone framework to protect location privacy of mobile users traveling on road networks. It makes three original contributions. First, we provide the formal analysis on the vulnerabilities of directly applying theoretical rectangle mix-zones to road networks in terms of anonymization effectiveness and resilience to timing and transition attacks. Second, we develop a suite of road network mix- zone construction methods that effectively consider the above mentioned factors to provide higher level of resilience to timing and transition attacks, and yield a specified lower-bound on the level of anonymity. Third, we present a set of mix-zone placement algorithms that identify the best set of road intersections for mix-zone placement considering the road network topology, user mobility patterns and road characteristics. We evaluate the MobiMix approach through extensive experiments conducted on traces produced by GTMobiSim on different scales of geographic maps. Our experiments show that MobiMix offers high level of anonymity and high level of resilience to timing and transition attacks, compared to existing mix-zone approaches.

Cost-effective Resource Provisioning for MapReduce in a Cloud

Balaji Palanisamy, Aameek Singh, Ling Liu

IEEE Transactions on Parallel and Distributed Systems (TPDS), May 2014.

This paper presents a new MapReduce cloud service model, Cura, for provisioning cost-effective MapReduce services in a cloud. In contrast to existing MapReduce cloud services such as a generic compute cloud or a dedicated MapReduce cloud, Cura has a number of unique benefits. Firstly, Cura is designed to provide a cost- effective solution to efficiently handle MapReduce production workloads that have a significant amount of interactive jobs. Secondly, unlike existing services that require customers to decide the resources to be used for the jobs, Cura leverages MapReduce profiling to automatically create the best cluster configuration for the jobs. While the existing models allow only a per-job resource optimization for the jobs, Cura implements a globally efficient resource allocation scheme that significantly reduces the resource usage cost in the cloud. Thirdly, Cura leverages unique optimization opportunities when dealing with workloads that can withstand some slack. By effectively multiplexing the available cloud resources among the jobs based on the job requirements, Cura achieves significantly lower resource usage costs for the jobs. Cura's core resource management schemes include cost-aware resource provisioning, VM-aware scheduling and online virtual machine reconfiguration. Our experimental results using Facebook-like workload traces show that our techniques lead to more than 80% reduction in the cloud compute infrastructure cost with up to 65% reduction in job response times.

continued on pg. 18

Recent Publications

continued from pg. 17

Aggregation and Degradation in JetStream: Streaming Analytics in the Wide Area

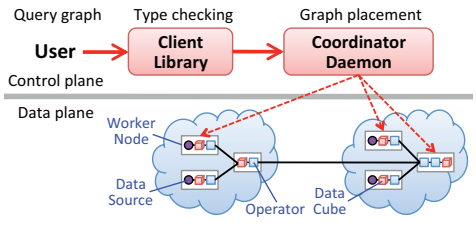
Ariel Rabkin, Matvey Arye, Siddhartha Sen, Vivek S. Pai, Michael J. Freedman

11th USENIX Symposium on Networked Systems Design and Implementation (NSDI'14), April 2014.

We present JetStream, a system that allows real-time analysis of large, widely-distributed changing data sets. Traditional approaches to distributed analytics require users to specify in advance which data is to be backhauled to a central location for analysis. This is a poor match for domains where available bandwidth is scarce and it is infeasible to collect all potentially useful data.

JetStream addresses bandwidth limits in two ways, both of which are explicit in the programming model. The system incorporates structured storage in the form of OLAP data cubes, so data can be stored for analysis near where it is generated. Using cubes, queries can aggregate data in ways and locations of their choosing. The system also includes adaptive filtering and other transformations that adjusts data quality to match available bandwidth. Many bandwidth-saving transformations are possible; we discuss which are appropriate for which data and how they can best be combined.

We implemented a range of analytic queries on web request logs and image data. Queries could be expressed in a few lines of code. Using structured storage on source nodes conserved network bandwidth by allowing data to be collected only when needed to fulfil queries. Our adaptive control mechanisms are responsive enough to keep end-to-end latency within a few seconds, even when available bandwidth drops by a factor of two, and are flexible enough to express practical policies.



JetStream's high-level architecture. Users define query graphs with operators and cubes. A coordinator deploys the graph to worker nodes.

From Application Requests to Virtual IOPs: Provisioned Key-value Storage with Libra

David Shue, Michael J. Freedman

Proceedings of the European Conference on Computer Systems (EuroSys '14), April 2014.

Achieving predictable performance in shared cloud storage services is hard. Tenants want reservations in terms of system-wide application-level throughput, but the provider must ultimately deal with low-level IO resources at each storage node where contention arises. Such a guarantee has thus proven elusive, due to the complexities inherent to modern storage stacks: non-uniform IO amplification, unpredictable IO interference, and non-linear IO performance.

This paper presents Libra, a local IO scheduling framework designed for a shared SSD-backed key-value storage system. Libra guarantees per-tenant application-request throughput while achieving high utilization. To accomplish this, Libra leverages two techniques. First, Libra tracks the IO resource consumption of a tenant's application-level requests across complex storage stack interactions, down to low-level IO operations. This allows Libra to allocate per-tenant IO resources for achieving app-request reservations based on their dynamic IO usage profile. Second, Libra uses a disk-IO cost model based on virtual IO operations (VOP) that captures the non-linear relationship between SSD IO bandwidth and IO operation (IOP) throughput. Using VOPs, Libra can both account for the true cost of an IOP and determine the amount of provisionable IO resources available under IO interference.

An evaluation shows that Libra, when applied to a LevelDB-based prototype with SSD-backed storage, satisfies tenant app-request reservations and achieves accurate low-level VOP allocations over a range of workloads, while still supporting high utilization.

So, You Want to Trace Your Distributed System? Key Design Insights from Years of Practical Experience

Raja R. Sambasivan, Rodrigo Fonseca, Ilari Shafer, Gregory R. Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-14-102, April 2014.

End-to-end tracing captures the workflow of causally-related activity (e.g., work done to process a request)within and among the components of a distributed system. As distributed systems grow in scale and complexity, such tracing is becoming a critical tool for management tasks like diagnosis and resource accounting. Drawing upon our experiences building and using end-to-end tracing infrastructures, this paper distills the key design axes that dictate trace utility for important use cases. Developing tracing infrastructures without explicitly understanding these axes and choices for them will likely result in infrastructures that are not useful for their intended purposes. In addition to identifying the design axes, this paper identifies good design choices for various tracing use cases, contrasts them to choices made by previous tracing implementations, and shows where prior implementations fall short. It also identifies remaining challenges on the path to making tracing an integral part of distributed system design.

Bounding Memory Interference Delay in COTS-based Multi-Core Systems

Hyoseung Kim, Dionisio de Niz, Bjorn Andersson, Mark Klein, Onur Mutlu, Ragunathan (Raj) Rajkumar

Proceedings of the 20th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'14), April 2014.

In commercial-off-the-shelf (COTS) multi-core systems, a task running on

one core can be delayed by other tasks running simultaneously on other cores due to interference in the shared DRAM main memory. Such memory interference delay can be large and highly variable, thereby posing a significant challenge for the design of predictable real-time systems. In this paper, we present techniques to provide a tight upper bound on the worst-case memory interference in a COTS-based multi-core system. We explicitly model the major resources in the DRAM system, including banks, buses and the memory controller. By considering their timing characteristics, we analyze the worst-case memory interference delay imposed on a task by other tasks running in parallel. To the best of our knowledge, this is the first work bounding the request re-ordering effect of COTS memory controllers. Our work also enables the quantification of the extent by which memory interference can be reduced by partitioning DRAM banks. We evaluate our approach on a commodity multi-core platform running Linux/RK. Experimental results show that our approach provides an upper bound very close to our measured worst-case interference.

Outsourcing Key-Value Stores with Verifiable Data Freshness

Yuzhe Tang, Ting Wang, Xin Hu, Reiner Sailer, Peter Pietzuch, Ling Liu

The 30th IEEE International Conference on Data Engineering (IEEE ICDE'14), April 2014.

In the age of big data, key-value data updated by intensive write streams is increasingly common, e.g., in social event streams. To serve such data in a cost-effective manner, a popular new paradigm is to outsource it to the cloud and store it in a scalable key-value store while serving a large user base. Due to the limited trust in third-party cloud infrastructures, data owners have to sign the data stream so that the data users can verify the authenticity of query results from the cloud. In this paper, we address the problem of verifiable freshness for multi-version key-value data. We propose a memory-resident digest structure that utilizes limited memory effectively and can have efficient verification performance. The proposed structure is named INCBM-TREE because it can INCrementally build a Bloom filter-embedded Merkle TREE. We have demonstrated the superior performance of verification under small memory footprints for signing, which is typical in an outsourcing scenario where data owners and users have limited resources.

MICA: A Holistic Approach to Near-Line-Rate In-Memory Key-Value Caching on General-Purpose Hardware

Hyeontaek Lim, Dongsu Han, David G. Andersen, Michael E. Kaminsky

11th USENIX Symposium on Networked Systems Design and Implementation (NSDI'14), April 2014.

MICA is a scalable in-memory key-value store that handles 65.6 to 76.9 million key-value operations per second using a single general-purpose multi-core system. MICA is over 4–13.5x faster than current state-of-the-art systems, while providing consistently high throughput over a variety of mixed read and write workloads.

MICA takes a holistic approach that encompasses all aspects of request handling, including parallel data access, network request handling, and data structure design, but makes unconventional choices in each of the three domains. First, MICA optimizes for multi-core architectures by enabling parallel access to partitioned data. Second, for efficient parallel data access, MICA maps client requests directly to specific CPU cores at the server NIC level by using client-supplied information and adopts a light-weight networking stack that bypasses the kernel. Finally, MICA's new data structures—circular logs, lossy concurrent hash indexes, and bulk chaining—handle both read- and write-intensive workloads at low overhead.

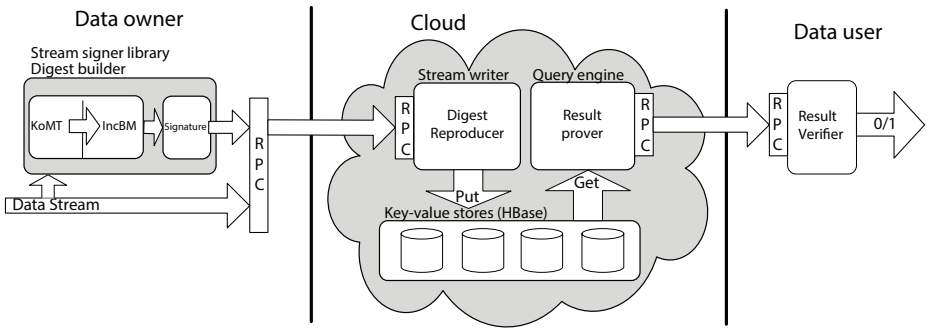
GRASS: Trimming Stragglers in Approximation Analytics

Ganesh Ananthanarayanan, Michael Chien-Chun Hung, Xiaoqi Ren, Ion Stoica, Adam Wierman, Minlan Yu

11th USENIX Symposium on Networked Systems Design and Implementation (NSDI'14), April 2014.

In big data analytics timely results, even if based on only part of the data, are often good enough. For this reason, approximation jobs, which have deadline or error bounds and require only a subset of their tasks to complete, are projected to dominate big data workloads. Straggler tasks are an important hurdle when designing approximate data analytic frameworks, and the widely adopted approach to deal with them is speculative execution. In this paper, we present GRASS, which carefully uses speculation to mitigate the impact of stragglers in approximation jobs. The design of GRASS is based on first principles analysis of the impact of speculative copies. GRASS delicately balances immediacy of improving the approximation goal with the long term implications of using extra resources for speculation. Evaluations with production workloads from Facebook and Microsoft Bing in an EC2 cluster of 200

continued on pg. 20



Outsourced system architecture based on HBase

Recent Publications

continued from pg. 19

nodes shows that GRASS increases accuracy of deadline-bound jobs by 47% and speeds up error-bound jobs by 38%. GRASS's design also speeds up exact computations, making it a unified solution for straggler mitigation.

GraphX: Unifying Data-Parallel and Graph-Parallel Analytics

Reynold Xin, Dan Crankshaw, Ankur Dave, Joseph Gonzalez, Michael Franklin, Ion Stoica

ArXiv, February 2014.

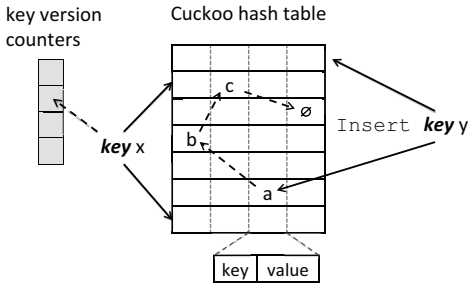
From social networks to language modeling, the growing scale and importance of graph data has driven the development of numerous new graph-parallel systems (e.g., Pregel, GraphLab). By restricting the computation that can be expressed and introducing new techniques to partition and distribute the graph, these systems can efficiently execute iterative graph algorithms orders of magnitude faster than more general data-parallel systems. However, the same restrictions that enable the performance gains also make it difficult to express many of the important stages in a typical graph-analytics pipeline: constructing the graph, modifying its structure, or expressing computation that spans multiple graphs. As a consequence, existing graph analytics pipelines compose graph-parallel and data-parallel systems using external storage systems, leading to extensive data movement and complicated programming model. To address these challenges we introduce GraphX, a distributed graph computation framework that unifies graph-parallel and data-parallel computation. GraphX provides a small, core set of graph-parallel operators expressive enough to implement the Pregel and PowerGraph abstractions, yet simple enough to be cast in relational algebra. GraphX uses a collection of query optimization techniques such as automatic join rewrites to efficiently implement these graph-parallel operators. We evaluate GraphX on real-world graphs and workloads and demonstrate that GraphX achieves comparable performance as specialized graph computation systems, while outperforming them in end-to-end graph pipelines. Moreover, GraphX achieves a balance between expressiveness, performance, and ease of use.

Algorithmic Improvements for Fast Concurrent Cuckoo Hashing

Xiaozhou Li, David G. Andersen, Michael A. Kaminsky, Michael J. Freedman

Proceedings of the European Conference on Computer Systems (EuroSys '14), April 2014.

Fast concurrent hash tables are an increasingly important building block as we scale systems to greater numbers of cores and threads. This paper presents the design, implementation, and evaluation of a high-throughput and memory-efficient concurrent hash table that supports multiple readers and writers. The design arises from careful attention to systems-level optimizations such as minimizing critical section length and reducing interprocessor coherence traffic through algorithm re-engineering. As part of the architectural basis for this engineering, we include a discussion of our experience and results adopting Intel's recent hardware transactional memory (HTM) support to this critical building block. We find that naively allowing concurrent access using a coarse-grained lock on existing data structures reduces overall performance with more threads. While HTM mitigates this slowdown somewhat, it does not eliminate it. Algorithmic optimizations that benefit both HTM and designs for fine-grained locking are needed to achieve high performance.



Cuckoo hash table overview: Each key is mapped to 2 buckets by hash functions and associated with 1 version counter. ∅ represents an empty slot. "a→b→c→∅" is a cuckoo path to make one bucket available to insert key y.

Our performance results demonstrate that our new hash table design—based around optimistic cuckoo hashing—outperforms other optimized concurrent hash tables by up to 2.5x for write-heavy workloads, even while using substantially less memory for small key-value items. On a 16-core machine, our hash table executes almost 40 million insert and more than 70 million lookup operations per second.

Personal Clouds: Sharing and Integrating Networked Resources to Enhance End User Experiences

Minsung Jang, Karsten Schwan, Ketan Bhardwaj, Ada Gavrilovska, Adhyas Avasthi

The 33rd IEEE International Conference on Computing Communications (Infocom), April 2014.

End user experiences on mobile devices with their rich sets of sensors are constrained by limited device battery lives and restricted form factors, as well as by the 'scope' of the data available locally. The 'Personal Cloud' distributed software abstractions address these issues by enhancing the capabilities of a mobile device via seamless use of both nearby and remote cloud resources. In contrast to vendor-specific, middleware-based cloud solutions, Personal Cloud instances are created at hypervisor-level, to create for each end user the federation of networked resources best suited for the current environment and use. Specifically, the Cirrostratus extensions of the Xen hypervisor can federate a user's networked resources to establish a personal execution environment, governed by policies that go beyond evaluating network connectivity to also consider device ownership and access rights, the latter managed in a secure fashion via standard Social Network Services. Experimental evaluations with both Linux- and Android-based devices, and using Facebook as the SNS, show the approach capable of substantially augmenting a device's innate capabilities, improving application performance and the effective functionality seen by end users.

Recent Publications

Efficient Instrumentation of GPGPU Programs using Information Flow Analysis and Symbolic Execution

Naila Farooqui, Karsten Schwan, Sudhakar Yalamanchili

Proceedings of Seventh Workshop on General-Purpose Computation on Graphics Processing Units (GPGPU-7), March 2014.

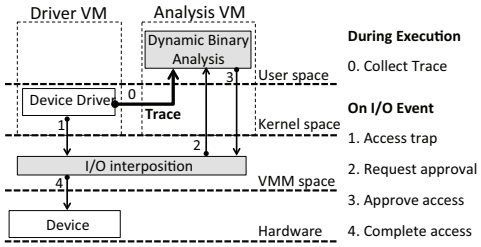
Dynamic instrumentation of GPGPU binaries makes possible real-time introspection methods for performance debugging, correctness checks, workload characterization, and runtime optimization. Such instrumentation involves inserting code at the instruction level of an application, while the application is running, thereby able to accurately profile data-dependent application behavior. Runtime overheads seen from instrumentation, however, can obviate its utility. This paper shows how a combination of information flow analysis and symbolic execution can be used to alleviate these overheads. The methods and their effectiveness are demonstrated for a variety of GPGPU codes written in OpenCL that run on AMD GPU target backends. Kernels that can be analyzed entirely via symbolic execution need not be instrumented, thus eliminating kernel runtime overheads altogether. For the remaining GPU kernels, our results show 5-38% improvements in kernel runtime overheads.

Guardrail: A High Fidelity Approach to Protecting Hardware Devices from Buggy Drivers

Olatunji Ruwase, Phillip B. Gibbons, Michael A. Kozuch, Todd Mowry

19th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS'14), March 2014.

Device drivers are an Achilles' heel of modern commodity operating systems, accounting for far too many system failures. Previous work on driver reliability has focused on protecting the kernel from unsafe driver side- effects



System architecture of Guardrail.

by interposing an invariant-checking layer at the driver interface, but otherwise treating the driver as a black box. In this paper, we propose and evaluate Guardrail, which is a more powerful framework for run-time driver analysis that performs decoupled, instruction-grain dynamic correctness checking on arbitrary kernel-mode drivers as they execute, thereby enabling the system to detect and mitigate more challenging correctness bugs (e.g., data races, uninitialized memory accesses) that cannot be detected by today's fault isolation techniques. Our evaluation of Guardrail shows that it can find serious data races, memory faults, and DMA faults in native Linux drivers that required fixes, including previously unknown bugs. Also, with hardware logging support, Guardrail can be used for online protection of persistent device state from driver bugs with at most 10% overhead on the end-to-end performance of most standard I/O workloads.

ParallelJS: An Execution Framework for Javascript on Heterogeneous Systems

J. Wang, N. Rubin, S. Yalamanchili

Proceedings of Seventh Workshop on General-Purpose Computation on Graphics Processing Units (GPGPU-7), March 2014.

JavaScript has been recognized as one of the most widely used script languages. Optimizations of JavaScript engines on mainstream web browsers enable efficient execution of JavaScript programs on CPUs. However, running JavaScript applications on emerging heterogeneous architectures that feature massively parallel hardware such as GPUs has not been well studied. This paper proposes a framework for flexible mapping of JavaScript onto heterogeneous systems that have both CPUs and GPUs. The framework includes a front-end compiler, a construct library and a runtime system. JavaScript programs written with high-level constructs are compiled to GPU binary code and scheduled to GPUs by the runtime. Experiments show that the proposed framework achieves up to 26.8x speedup executing JavaScript applications on parallel GPUs over a mainstream web browser that runs on CPUs.

Red Fox: An Execution Environment for Accelerating Relational Queries using GPUs

H. Wu, G. Diamos, T. Sheard, M. Aref, S. Yalamanchili

IEEE/ACM International Symposium on Code Generation and Optimization (CGO'14), February 2014.

Modern enterprise applications represent an emergent application arena that requires the processing of queries and computations over massive amounts of data. Large-scale, multi-GPU cluster systems potentially present a vehicle for major improvements in throughput and consequently overall performance. However, throughput improvement using GPUs is challenged by the distinctive memory and computational characteristics of Relational Algebra (RA) operators that are central to queries for answering business questions. This paper introduces the design, implementation, and evaluation of Red Fox, a compiler and runtime infrastructure for executing relational queries on GPUs. Red Fox is comprised of i) a language front-end for LogiQL which is a commercial query language, ii) an RA to GPU compiler, iii) optimized GPU implementation of RA operators, and iv) a supporting runtime. We report the performance on the full set of industry standard TPC-H queries on a single node GPU. Compared with a commercial LogiQL system implementation optimized for a state of art CPU machine, Red Fox on average is 6.48x faster including

continued on pg. 22

Recent Publications

continued from pg. 21

PCIe transfer time. We point out key bottlenecks, propose potential solutions, and analyze the GPU implementation of these queries. To the best of our knowledge, this is the first reported end-to-end compilation and execution infrastructure that supports the full set of TPC-H queries on commodity GPUs.

Agentless Cloud-wide Streaming of Guest File System Updates

Wolfgang Richter, Canturk Isci, Jan Harkes, Benjamin Gilbert, Vasanth Bala, Mahadev Satyanarayanan

The Second IEEE Conference on Cloud Engineering (IC2E'14), March 2014.

We propose a non-intrusive approach for monitoring virtual machines (VMs) in the cloud. At the core of this approach is a mechanism for selective real-time monitoring of guest file updates within VM instances. This mechanism is agentless, requiring no guest VM support. It has low virtual I/O overhead, low latency for emitting file updates, and a scalable design. Its central design principle is distributed streaming of file updates inferred from introspected disk sector writes. The mechanism, called DS-VMI, enables many system administration tasks that involve monitoring files to be performed outside VMs.

Reducing the Cost of Persistence for Nonvolatile Heaps in End User Devices

Sudarsun Kannan, Ada Gavrilovska, Karsten Schwan

Proceedings of the 20th International Symposium on High-Performance Computer Architecture (HPCA'14), February 2014.

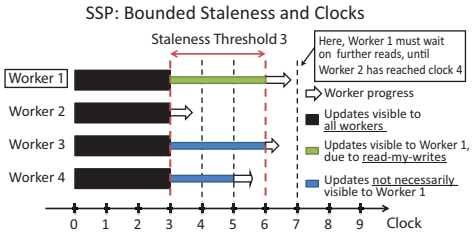
This paper explores the performance implications of using future byte addressable non-volatile memory (NVM) like PCM in end client devices. We explore how to obtain dual benefits — increased capacity and faster persistence — with low overhead and cost. Specifically, while increasing memory capacity can be gained by treating NVM as virtual memory, its use of persistent data storage incurs high consistency (frequent cache flushes) and durability (logging for failure) overheads, referred to as 'persistence cost'. These not only affect the applications causing them, but also other applications relying on the same cache and/or memory hierarchy. This paper analyzes and quantifies in detail the performance overheads of persistence, which include (1) the aforementioned cache interference as well as (2) memory allocator overheads, and finally, (3) durability costs due to logging. Novel solutions to overcome such overheads include (1) a page contiguity algorithm that reduces interference-related cache misses, (2) a cache efficient NVM write aware memory allocator that reduces cache line flushes of allocator state by 8X, and (3) hybrid logging that reduces durability overheads substantially. With these solutions, experimental evaluations with different end user applications and SPEC2006 benchmarks show up to 12% reductions in cache misses, thereby reducing the total number of NVM writes.

More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server

Q. Ho, J. Cipar, H. Cui, S. Lee, J. Kim, P. Gibbons, G. Gibson, G. Ganger, and E. Xing

Neural Information Processing Systems Conference (NIPS'13), December 2013.

We propose a parameter server system for distributed ML, which follows a Stale Synchronous Parallel (SSP) model of computation that maximizes the time computational workers spend doing useful work on ML algorithms, while still providing correctness guarantees. The parameter server provides an easy-to-use shared interface for read/write access to an ML model's values (parameters and variables), and the SSP model allows distributed workers to read older, stale versions of these values from a local cache, instead of waiting to get them from a central storage. This significantly increases the proportion of time workers spend computing, as opposed to waiting. Furthermore, the SSP model



Bounded Staleness under the SSP Model

ensures ML algorithm correctness by limiting the maximum age of the stale values. We provide a proof of correctness under SSP, as well as empirical results demonstrating that the SSP model achieves faster algorithm convergence on several different ML problems, compared to fully-synchronous and asynchronous schemes.

SpringFS: Bridging Agility and Performance in Elastic Distributed Storage

Lianghong Xu, James Cipar, Elie Krevat, Alexey Tumanov, Nitin Gupta, Michael A. Kozuch, Gregory R. Ganger

12th USENIX Conference on File and Storage Technologies (FAST '14), February 2014.

Elastic storage systems can be expanded or contracted to meet current demand, allowing servers to be turned off or used for other tasks. However, the usefulness of an elastic distributed storage system is limited by its agility: how quickly it can increase or decrease its number of servers. Due to the large amount of data they must migrate during elastic resizing, state-of-the-art designs usually have to make painful tradeoffs among performance, elasticity and agility. This paper describes an elastic storage system, called SpringFS, that can quickly change its number of active servers, while retaining elasticity and performance goals. SpringFS uses a novel technique, termed bounded write offloading, that restricts the set of servers where writes to overloaded servers are redirected.

This technique, combined with the read offloading and passive migration policies used in SpringFS, minimizes

the work needed before deactivation or activation of servers. Analysis of real-world traces from Hadoop deployments at Facebook and various Cloudera customers and experiments with the SpringFS prototype confirm SpringFS's agility, show that it reduces the amount of data migrated for elastic resizing by up to two orders of magnitude, and show that it cuts the percentage of active servers required by 67–82%, outdoing state-of-the-art designs by 6–120%.

Parameter Server for Distributed Machine Learning

Mu Li, Li Zhou, Zichao Yang, Aaron Li, Fei Xia, David G. Andersen, Alexander Smola

Workshop on Big Learning: Advances in Algorithms and Data Management, with NIPS'13, December 2013.

We propose a parameter server framework to solve distributed machine learning problems. Both data and workload are distributed into client nodes, while server nodes maintain globally shared parameters, which are represented as sparse vectors and matrices. The framework manages asynchronous data communications between clients and servers. Flexible consistency models, elastic scalability and fault tolerance are supported by this framework. We present algorithms and theoretical analysis for challenging nonconvex and nonsmooth problems. To demonstrate the scalability of the proposed framework, we show experimental results on real data with billions of parameters.

QuiltView: Glass-Sourced Video for Google Maps Queries

Zhuo Chen, Wenlu Hu, Kiryong Ha, Jan Harkes, Benjamin Gilbert, Jason Hong, Asim Smailagic, Dan Siewiorek, Mahadev Satyanarayanan

The 15th International Workshop on Mobile Computing Systems and Applications (HotMobile'14), February 2014.

Effortless one-touch capture of video is a unique capability of wearable devices such as Google Glass. We use this capability to create a new type of crowd-sourced system in which users receive queries relevant to their current location and opt-in preferences. In response, they can send back live video snippets of their surroundings. A system of result caching, geolocation and query similarity detection shields users from being overwhelmed by a flood of queries.

Distributed Delayed Proximal Gradient Methods

Mu Li, Dave Andersen, Alex Smola

Workshop on OPT2013: Optimization for Machine Learning, with NIPS'13, December 2013.

We analyze distributed optimization algorithms where parts of data and variables are distributed over several machines and synchronization occurs asynchronously. We prove convergence for the general case of a nonconvex objective plus a convex and possibly nonsmooth penalty. We demonstrate two challenging applications, l_1-regularized logistic regression and reconstruction ICA, and present experiments on real datasets with billions of variables using both CPUs and GPUs.

ADAM: Genomics Formats and Processing Patterns for Cloud Scale Computing

Matt Massie, Frank Nothaft, Christopher Hartl, Christos Kozanitis, Andre Schumacher, Anthony D. Joseph, David A. Patterson

Berkeley Technical Report No. UCB/EECS-2013-207, December 2013.

Current genomics data formats and processing pipelines are not designed to scale well to large datasets. The current Sequence/Binary Alignment/Map (SAM/BAM) formats were intended for single node processing [18]. There have been attempts to adapt BAM to distributed computing environments, but they see limited scalability past eight nodes [22]. Additionally, due to the lack of an explicit data schema, there are well known incompatibilities between libraries that implement SAM/BAM/Variant Call Format (VCF) data access.

To address these problems, we introduce ADAM, a set of formats, APIs, and processing stage implementations for genomic data. ADAM is fully open source under the Apache 2 license, and is implemented on top of Avro and Parquet [5, 26] for data storage. Our reference pipeline is implemented on top of Spark, a high performance in-memory map-reduce system [32]. This combination provides the following advantages: 1) Avro provides explicit data schema access in C/C++/C#, Java/Scala, Python, php, and Ruby; 2) Parquet allows access by database systems like Impala and Shark; and 3) Spark improves performance through in-memory caching and reducing disk I/O.

Visualizing Request-flow Comparison to Aid Performance Diagnosis in Distributed Systems

Raja R. Sambasivan, Ilari Shafer, Michelle L. Mazurek, and Gregory R. Ganger

Transactions on Visualization and Computer Graphics (Proceedings Information Visualization 2013), vol. 19, no. 12, Dec. 2013.

Distributed systems are complex to develop and administer, and performance problem diagnosis is particularly challenging. When performance degrades, the problem might be in any of the system's many components or could be a result of poor interactions among them. Recent research efforts have created tools that automatically localize the problem to a small number of potential culprits, but research is needed to understand what visualization techniques work best for helping distributed systems developers understand and explore their results. This paper compares the relative merits of three well-known visualization approaches (side-by-side, diff, and animation) in the context of presenting the results of one proven automated localization technique called request-flow comparison. Via a 26-person user study, which included real distributed

continued on pg. 24

Recent Publications

continued from pg. 23

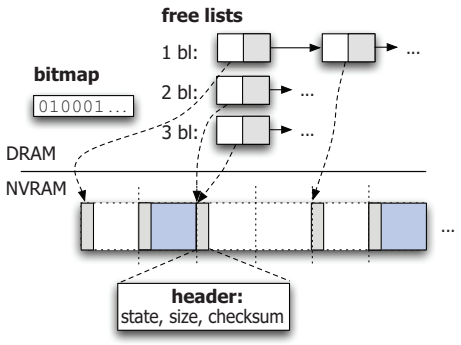
systems developers, we identify the unique benefits that each approach provides for different problem types and usage modes.

Consistent, Durable, and Safe Memory Management for Byte-addressable Non-Volatile Main Memory

Iulian Moraru, David G. Andersen, Michael Kaminsky, Parthasarathy Ranganathan, Niraj Tolia, Nathan Binkert

First ACM Conference on Timely Results in Operating Systems (TRIOS'13), with SOSP'13, November 2013.

This paper presents three building blocks for enabling the efficient and safe design of persistent data stores for emerging non-volatile memory technologies. Taking the fullest advantage of the low latency and high bandwidths of emerging memories such as phase change memory (PCM), spin torque, and memristor necessitates a serious look at placing these persistent storage technologies on the main memory bus. Doing so, however, introduces critical challenges of not sacrificing the data reliability and consistency that users demand from storage. This paper introduces techniques for (1) robust wear-aware memory allocation, (2) preventing of erroneous writes, and (3) consistency-preserving updates that are cache-efficient. We show through our evaluation that these techniques are efficiently implementable and effective by demonstrating a B+-tree implementation modified to make full use of our toolkit.



Memory allocator metadata example. Two of the total six basic memory blocks depicted in the diagram are allocated.

inTune: Coordinating Multicore Islands to Achieve Global Policy Objectives

Priyanka Tembey, Ada Gavriloska, Karsten Schwan

First ACM Conference on Timely Results in Operating Systems (TRIOS'13), with SOSP'13, November 2013.

Multicore platforms are moving from small numbers of homogeneous cores to 'scale out' designs with multiple tiles or 'islands' of cores residing on a single chip, each with different resources and potentially controlled by their own resource managers. Applications running on such machines, however, operate across multiple such resource islands, and this also holds for global properties like platform power caps. The inTune software architecture meets the consequent need to support platform-level application requirements and properties. It (i) provides the base coordination abstractions needed for realizing platform-global resource management and (ii) offers management overlays that make it easy to implement diverse per-application and platform-centric management policies. A Xen hypervisor-level implementation of inTune supports policies that can (i) pro-actively prepare for increased or decreased resource usage when the inter-island dependencies of applications are known, or (ii) re-actively respond to monitored overloads, threshold violations or similar. Experimental evaluations on a larger-scale multi-core platform demonstrate that its use leads to notable performance and resource utilization gains: such as a reduction in the variability across request response times for a three-tier web server by up to 40%, and completion time gains of 15% for parallel benchmarks.

NVM Heaps for Accelerating Browser-based Applications

Sudarsan Kannan, Ada Gavrilvoska, Karsten Schwan

Workshop on Interactions of NVM/Flash with Operating-Systems and Workloads (INFLOW'13), with SOSP'13, November 2013.

The growth in browser-based computations is raising the need for efficient local storage for browser-based applications. A standard approach to control how such applications access and manipulate the underlying platform resources, is to run in-browser applications in a sandbox environment. Sandboxing works by static code analysis and system call interception, and as a result, the performance of browser applications making frequent I/O calls can be severely impacted. To address this, we explore the utility of next generation non-volatile memories (NVM) in client platforms. By using NVM as virtual memory, and integrating NVM support for browser applications with byte addressable I/O interfaces, our approach shows up to 3.5x reduction in sandboxing cost and around 3x reduction in serialization overheads for browser based applications, and improved application performance.

Efficient Data Partitioning Model for Heterogeneous Graphs in the Cloud

Kisung Lee, Ling Liu

IEEE international Conference for High Performance Computing, Networking, Storage and Analysis (SC2013), November 2013.

As the size and variety of information networks continue to grow in many scientific and engineering domains, we witness a growing demand for efficient processing of large heterogeneous graphs using a cluster of compute nodes in the Cloud. One open issue is how to effectively partition a large graph to process complex graph operations efficiently. In this paper, we present VB-Partitioner -- a distributed data partitioning model and algorithms for efficient processing of graph op-

erations over large-scale graphs in the Cloud. Our VB-Partitioner has three salient features. First, it introduces vertex blocks (VBs) and extended vertex blocks (EVBs) as the building blocks for semantic partitioning of large graphs. Second, VB-Partitioner utilizes vertex block grouping algorithms to place those vertex blocks that have high correlation in graph structure into the same partition. Third, VB-Partitioner employs a VB-partition guided query partitioning model to speed up the parallel processing of graph pattern queries by reducing the amount of inter-partition query processing. We conduct extensive experiments on several real-world graphs with millions of vertices and billions of edges. Our results show that VB-Partitioner significantly outperforms the popular random block-based data partitioner in terms of query latency and scalability over large-scale graphs.

There Is More Consensus in Egalitarian Parliaments

Iulian Moraru, David G. Andersen, Michael Kaminsky

24th ACM Symposium on Operating Systems Principles (SOSP'13), November 2013.

This paper describes the design and implementation of Egalitarian Paxos (EPaxos), a new distributed consensus algorithm based on Paxos. EPaxos achieves three goals: (1) optimal commit latency in the wide-area when tolerating one and two failures, under realistic conditions; (2) uniform load balancing across all replicas (thus achieving high throughput); and (3) graceful performance degradation when replicas are slow or crash.

Egalitarian Paxos is to our knowledge the first protocol to achieve the previously stated goals efficiently—that is, requiring only a simple majority of replicas to be non-faulty, using a number of messages linear in the number of replicas to choose a command, and committing commands after just one communication round (one round trip) in the common case or after at most two rounds in any case. We prove Egalitarian Paxos's properties theoretically and demonstrate its advantages empirically through an implementation running on Amazon EC2.

PARROT: A Practical Runtime for Deterministic, Stable, and Reliable Threads

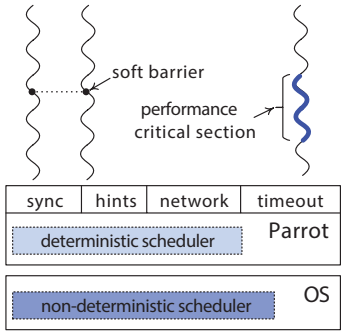
Heming Cui, Jiri Simsa, Yi-Hong Lin, Hao Li, Ben Blum, Xinan Xu, Junfeng Yang, Garth A. Gibson, Randal E. Bryant

24th ACM Symposium on Operating Systems Principles (SOSP'13), November 2013.

Multi-threaded programs are hard to get right. A key reason is that the contract between developers and runtimes grants exponentially many schedules to the runtimes. We present PARROT, a simple, practical runtime with a new contract to developers. By default, it orders thread synchronizations in the well-defined round-robin order, vastly reducing schedules to provide determinism (more precisely, deterministic synchronizations) and stability (i.e., robustness against input or code perturbations, a more useful property than determinism). When default schedules are slow, it allows developers to write intuitive performance hints in their code to switch or add schedules for speed. We believe this "meet in the middle" contract eases writing correct, efficient programs.

We further present an ecosystem formed by integrating PARROT with a model checker called DBUG. This ecosystem is more effective than either system alone: DBUG checks the schedules that matter to PARROT, and PARROT greatly increases the coverage of DBUG.

Results on a diverse set of 108 programs, roughly 10× more than any prior evaluation, show that PARROT is easy to use (averaging 1.2 lines of hints per program); achieves low overhead (6.9% for 55 real-world programs and 12.7% for all 108 programs), 10× better than two prior systems; scales well to the maximum allowed cores on a 24-core server and to different scales/types of workloads; and increases DBUG's coverage by



PARROT architecture.

10^6 – 10^{19734} for 56 programs. PARROT's source code, entire benchmark suite, and raw results are available at github.com/columbia/smt-mc.

The Role of Cloudlets in Hostile Environments

M. Satyanarayanan, G. Lewis, E. Morris, S. Simanta, J. Boleng, K. Ha

IEEE Pervasive Computing (PerCom'13), October 2013.

The convergence of mobile computing and cloud computing is predicated on a reliable, high-bandwidth, end-to-end network, which is difficult to guarantee in hostile environments. However, virtual-machine-based cloudlets located in close proximity to associated mobile devices can overcome this deep-rooted problem.

Sparrow: Distributed, Low Latency Scheduling

Kay Ousterhout, Patrick Wendell, Matei Zaharia, Ion Stoica

24th ACM Symposium on Operating Systems Principles (SOSP'13), November 2013.

Large-scale data analytics frameworks are shifting towards shorter task durations and larger degrees of parallelism to provide low latency. Scheduling highly parallel jobs that complete in hundreds of milliseconds poses a major challenge for task schedulers, which will need to schedule millions of tasks per second on appropriate machines while offering millisecond-level latency

continued on pg. 26

Recent Publications

continued from pg. 25

and high availability. We demonstrate that a decentralized, randomized sampling approach provides near-optimal performance while avoiding the throughput and availability limitations of a centralized design. We implement and deploy our scheduler, Sparrow, on a 110-machine cluster and demonstrate that Sparrow performs within 12% of an ideal scheduler.

Discretized Streams: Fault-Tolerant Streaming Computation at Scale

Matei Zaharia, Tathagata Das, Haoyuan Li, Timothy Hunter, Scott Shenker, Ion Stoica

24th ACM Symposium on Operating Systems Principles (SOSP'13), November 2013.

Large-scale data analytics frameworks are shifting towards shorter task durations and larger degrees of parallelism to provide low latency. Scheduling highly parallel jobs that complete in hundreds of milliseconds poses a major challenge for task schedulers, which will need to schedule millions of tasks per second on appropriate machines while offering millisecond-level latency and high availability. We demonstrate that a decentralized, randomized sampling approach provides near-optimal performance while avoiding the throughput and availability limitations of a centralized design. We implement and deploy our scheduler, Sparrow, on a 110-machine cluster and demonstrate that Sparrow performs within 12% of an ideal scheduler.

Memory-Efficient GroupBy-Aggregate using Compressed Buffer Trees

Hrishikesh Amur, Wolfgang Richter, David G. Andersen, Michael Kaminsky, Karsten Schwan, Athula Balanachandran, Erik Zawadzki

4th ACM Symposium on Cloud Computing (SOCC'13), October 2013.

The rapid growth of fast analytics systems, that require data processing in memory, makes memory capacity an increasingly-precious resource. This paper introduces a new compressed data structure called a Compressed Buffer Tree (CBT). Using a combination of techniques including buffering, compression, and serialization, CBTs improve the memory efficiency and performance of the GroupBy-Aggregate abstraction that forms the basis of not only batch-processing models like MapReduce, but recent fast analytics systems too. For streaming workloads, aggregation using the CBT uses 21-42% less memory than using Google SparseHash with up to 16% better throughput. The CBT is also compared to batch-mode aggregators in MapReduce runtimes such as Phoenix++ and Metis and consumes 4 and 5 less memory with 1.5-2 and 3-4 more performance respectively.

vTube: Efficient Streaming of Virtual Appliances Over Last-Mile Networks

Yoshihisa Abe, Roxana Geambasu, Kaustubh Joshi, H. Andres Lagar-Cavilla, Mahadev Satyanarayanan

4th ACM Symposium on Cloud Computing (SOCC'13), October 2013.

Cloud-sourced virtual appliances (VAs) have been touted as powerful solutions for many software maintenance, mobility, backward compatibility, and security challenges. In this paper, we ask whether it is possible to create a VA cloud service that supports fluid, interactive user experience even over mobile networks. More specifically, we wish to support a YouTube-like streaming service for executable content, such as games, interactive books, research artifacts, etc. Users should be able to post, browse through, and interact with executable content swiftly and without long interruptions. Intuitively, this seems impossible; the bandwidths, latencies, and costs of last-mile networks would be prohibitive given the sheer sizes of virtual machines! Yet, we show that a set of carefully crafted, novel prefetching and streaming techniques can bring this goal surprisingly close to reality. We show that vTube, a VA streaming system that incorporates our techniques, supports fluid interaction even in challenging network conditions, such as 4G LTE.

Hierarchical Scheduling for Diverse Datacenter Workloads

Arka Bhattacharya, Eric Friedman, Ali Ghodsi, Scott Shenker, Ion Stoica

4th ACM Symposium on Cloud Computing (SOCC'13), October 2013.

There has been a recent industrial effort to develop multi-resource hierarchical schedulers. However, the existing implementations have some shortcomings in that they might leave resources unallocated or starve certain jobs. This is because the multi-resource setting introduces new challenges for hierarchical scheduling policies. We provide an algorithm, which we implement in Hadoop, that generalizes the most commonly used multi-resource scheduler, DRF [1], to support hierarchies. Our evaluation shows that our proposed algorithm, H-DRF, avoids the starvation and resource inefficiencies of the existing open-source schedulers and outperforms slot scheduling.

An Infrastructure for Automating Large-scale Performance Studies and Data Processing

Deepal Jayasinghe, Josh Kimball, Tao Zhu, Siddharth Choudhary, and Calton Pu

IEEE Big Data Conference (IEEE BigData'13), October 2013.

The Cloud has enabled the computing model to shift from traditional data centers to publicly shared computing infrastructure; yet, applications leveraging this new computing model can experience performance and scalability issues, which arise from the hidden complexities of the cloud. The most reliable path for better understanding these complexities is an empirically based approach that relies on collecting data from a large number of performance studies. Armed with this performance data, we can understand what has happened, why it happened, and more importantly, predict what will happen in the future. However, this approach presents challenges itself,

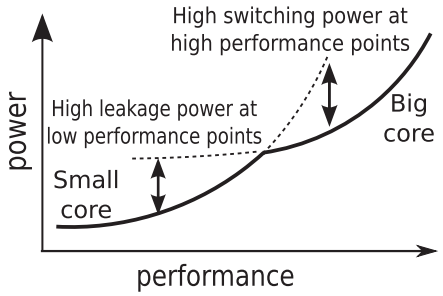
namely in the form of data management. We attempt to mitigate these data challenges by fully automating the performance measurement process. Concretely, we have developed an automated infrastructure, which reduces the complexity of the large-scale performance measurement process by generating all the necessary resources to conduct experiments, to collect and process data and to store and analyze data. In this paper, we focus on the performance data management aspect of our infrastructure.

Core Groups: System Abstractions for Extending the Dynamic Range of Client Devices using Heterogeneous Cores

Vishal Gupta, Paul Brett, David Koufaty, Dheeraj Reddy, Scott Hahn, Karsten Schwan, Ganapati Srinivasa

Elsevier Journal of Sustainable Computing (SUSCOM), 3(3), Selected papers from the 2012 IEEE International Green Computing Conference, September 2013.

Mobile devices and applications exhibit highly diverse behavior in their usage and power/performance requirements. In order to accommodate such diversity, this paper presents 'HeteroMates' system that uses heterogeneous processors to extend the dynamic power/performance range of client devices, i.e., offer both high performance and reduced power consumption. It proposes core group abstraction that groups a small number of heterogeneous cores to form a single execution unit. Group heterogeneity is exposed as multiple heterogeneity (H) states, an interface similar to the P-state interface already used for frequency scaling. Further, the core group abstraction is extended to a multicore group to allow multiple cores within a group to be active concurrently. Also demonstrated is the importance of 'uncore' power in total SoC power consumption and the need for uncore-aware operation and uncore power scalability when seeking to extend a platform's dynamic power/performance range using heterogeneity. Experimental evaluations use real-world client applications and a unique



Big cores are less efficient at low activity points, while small cores are less efficient at high activity points. Using a heterogeneous processor provides a wide dynamic power/performance range.

experimental testbed comprised of heterogeneous cores and a shared uncore component. Results show that HeteroMates can provide significant performance improvements while also lowering energy consumption for a diverse set of applications when compared to homogeneous processor configurations.

Performance Troubleshooting in Datacenters

Chengwei Wang, Soila Pertet Kavulya, Jiaqi Tan, Michael Kasick, Liting Hu, Mahendra Kutare, Priya Narasimham, Karsten Schwan, Rajeev Gandhi

Operating Systems Review, October 2013.

In the emerging cloud computing era, enterprise data centers host a plethora of web services and applications, including those for e-Commerce, distributed multimedia, and social networks, which jointly, serve many aspects of our daily lives and business. For such applications, lack of availability, reliability, or responsiveness can lead to extensive losses. For instance, on June 29th 2010, Amazon.com experienced three hours of intermittent performance problems as the normally reliable website took minutes to load items, and searches came back without product links. Customers were also unable to place orders. Based on their 2010 quarterly revenues, such downtime could cost Amazon up to $1.75 million per hour, thus making rapid problem resolution critical to its business. In another serious incident, on July 7th, 2010, DBS bank in Singapore suffered a 7-hour outage which crippled its Internet banking systems, and disrupted other consumer banking services, including automated teller machines, credit card and NETS payments. The cascading failure occurred due to a procedural error while replacing a faulty component in one of the bank's storage systems that was connected to its main computers.

Who Is Your Neighbor: Net I/O Performance Interference in Virtualized Clouds

Xing Pu, Ling Liu, Yiduo Mei, Sankaran Sivathanu, Younggyun Koh, Calton Pu, Yuanda Cao

IEEE Transactions on Services Computing, Vol. 6, No. 3, July-September, 2013.

User-perceived performance continues to be the most important QoS indicator in cloud-based data centers today. Effective allocation of virtual machines (VMs) to handle both CPU intensive and I/O intensive workloads is a crucial performance management capability in virtualized clouds. Although a fair amount of researches have dedicated to measuring and scheduling jobs among VMs, there still lacks of in-depth understanding of performance factors that impact the efficiency and effectiveness of resource multiplexing and scheduling among VMs. In this paper, we present the experimental research on performance interference in parallel processing of CPU-intensive and network-intensive workloads on Xen virtual machine monitor (VMM). Based on our study, we conclude with five key findings which are critical for effective performance management and tuning in virtualized clouds. First, co-locating network-intensive workloads in isolated VMs incurs high overheads of switches and events in Dom0 and VMM. Second, colocating CPU-intensive workloads in isolated VMs incurs high CPU contention due to fast I/O processing in I/O channel. Third, running CPU-intensive and network-intensive workloads in conjunction incurs the least resource contention, delivering higher aggregate performance. Fourth, performance of

continued on pg. 28

Recent Publications

continued from pg. 27

network-intensive workload is insensitive to CPU assignment among VMs, whereas adaptive CPU assignment among VMs is critical to CPU-intensive workload. The more CPUs pinned on Dom0 the worse performance is achieved by CPU-intensive workload. Last, due to fast I/O processing in I/O channel, limitation on grant table is a potential bottleneck in Xen. We argue that identifying the factors that impact the total demand of exchanged memory pages is important to the in-depth understanding of interference costs in Dom0 and VMM.

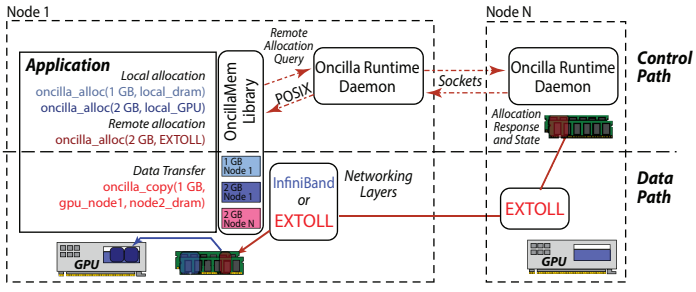
Oncilla: A GAS Runtime for Efficient Resource Allocation and Data Movement in Accelerated Clusters

J. Young, S. H. Shon, S. Yalamanchili, A. Merrit, K. Schwan, H. Froening

IEEE International Conference on Cluster Computing (Cluster'13), September 2013.

Accelerated and in-core implementations of Big Data applications typically require large amounts of host and accelerator memory as well as efficient mechanisms for transferring data to and from accelerators in heterogeneous clusters. Scheduling for heterogeneous CPU and GPU clusters has been investigated in depth in the high-performance computing (HPC) and cloud computing arenas, but there has been less emphasis on the management of cluster resource that is required to schedule applications across multiple nodes and devices. Previous approaches to address this resource management problem have focused on either using low-performance software layers or on adapting complex data movement techniques from the HPC arena, which reduces performance and creates barriers for migrating applications to new heterogeneous cluster architectures.

This work proposes a new system architecture for cluster resource allocation and data movement built around the concept of managed Global Address Spaces (GAS), or dynamically aggregated memory regions that span multiple nodes. We propose a software layer called Oncilla that uses a simple runtime and API to take advantage of non-coherent hardware support for GAS. The Oncilla runtime is evaluated using two different high-performance networks for microkernels representative of the TPC-H data warehousing benchmark, and this runtime enables a reduction in runtime of up to 81%, on average, when compared with standard disk-based data storage techniques. The use of the Oncilla API is also evaluated for a simple breadth-first search (BFS) benchmark to demonstrate how existing applications can incorporate support for managed GAS.



Oncilla Runtime and use of API

Other Interesting Papers by ISTC-CC Faculty

See http://www.istc-cc.cmu.edu/publications/index.shtml

Reducing the Sampling Complexity of Topic Models. Aaron Li, Amr Ahmed, Sujith Ravi, Alex Smola. Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'14), August 2014.

Activity-edge Centric Multi-label Classification for Mining Heterogeneous Information Networks. Yang Zhu, Ling Liu. Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'14), August 2014.

Warp-Aware Trace Scheduling for GPUs. James Jablin, Thomas Jablin, Onur Mutlu, and Maurice Herlihy, Proceedings of the 23rd ACM International Conference on Parallel Architectures and Compilation Techniques (PACT'14), August 2014.

DeSTM: Harnessing Determinism in STMs for Application Development. Kaushik Ravichandran, Ada Gavrilovska, Santosh Pande. Proceedings of the 23rd ACM International Conference on Parallel Architectures and Compilation Techniques (PACT'14), August 2014.

Rollback-Free Value Prediction with Approximate Memory Loads. Bradley Thwaites, Gennady Pekhimenko, Amir Yazdanbakhsh, Girish Mururu, Jongse Park, Hadi Esmaeilzadeh, Onur Mutlu, Todd C. Mowry. Proceedings of the 23rd ACM International Conference on Parallel Architectures and Compilation Techniques (PACT'14), August 2014.

XIA: Architecting a More Trustworthy and Evolvable Internet. D. Naylor, M. K. Mukerjee, P. Agyapong, R. Grandl, R. Kang, M. Machado, S. Brown, C.Doucette, H.-C. Hsiao, D.Han, T. Hyun-Jin Kim, H. Lim, C. Ovon, D. Zhou, S. Bum Lee, Y.-H. Lin, C. Stuart, D. Barrett, A. Akella, D. Andersen, J. Byers, L. Dabbish, M. Kaminsky, S. Kiesler, J. Peha, A. Perrig, S. Seshan, M. Sirbu, P. Steenkiste. ACM SIGCOMM Computing and Communications Review (CCR), 44(3), July 2014.

Context-Aware Cloud Service Selection Based On Comparison and Aggregation of User Subjective Assessment and Objective Performance Assessment. Lie Qu, Yan Wang, Mehmet A. Orgun, Ling Liu, Athman Bouguettaya. Proceedings of the 21st IEEE International Conference on Web Services (ICWS'14), June-July 2014.

Landslide Detection Service Based on Composition of Physical and Social information Services. Aibek Musaev, De Wang, Chien-An Cho, Calton Pu. Proceedings of the 21st IEEE International Conference on Web Services (ICWS'14), June-July 2014.

e-PPI: Searching Information Networks with Quantitative Privacy Guarantees. Yuzhe Tang, Ling Liu, Arun Iyengar, Kisung Lee, Qi Zhang. Proceedings of 34th IEEE International Conference on Distributed Computing Systems (ICDCS'14), June-July 2014.

The Impact of Software Resource Allocation on Consolidated n-Tier Applications. Jack Li, Qingyang Wang,

Recent Publications

Students, faculty and industry members of the ISTC-CC attend a poster session at the 3rd Annual Retreat.

Chien-An Lai, Junhee Park, Daisaku Yokoyama, Calton Pu. Proceedings of IEEE 7th Int. Conf. on Cloud Computing (Cloud'14), June-July 2014.

Experimental Analysis of Space-Bounded Schedulers. Harsha Vardhan Simhadri, Guy Blelloch, Jeremy T. Fineman, Phillip B. Gibbons, Aapo Kyrola. Proceedings of the 26th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'14), June 2014.

A Simple and Practical Linear-Work Parallel Algorithm for Connectivity. Julian Shun, Laxman Dhulipala, Guy Blelloch. Proceedings of the 26th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'14), June 2014.

Adaptive Delay-Tolerant Scheduling for Efficient Cellular and WiFi Usage. Ozlem Bilgir Yetim, Margaret Martonosi. Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'14), June 2014.

Memory Systems. Yoongu Kim, Onur Mutlu. Invited Book Chapter in Computing Handbook, Third Edition: Computer Science and Software Engineering, CRC Press, April 2014.

Outsourcing Key-Value Stores with Verifiable Data Freshness. Yuzhe Tang, Ting Wang, Xin Hu, Reiner Sailer, Peter Pietzuch, Ling Liu. Proceedings of the 30th IEEE International Conference on Data Engineering (ICDE'14), April 2014.

Efficient Instrumentation of GPGPU Applications using Information Flow Analysis and Symbolic Execution. Naila Farooqui, Karsten Schwan, Sudhakar Yalamanchili. Proceedings of Seventh Workshop on General-Purpose Computation on Graphics Processing Units (GPGPU '14), March 2014.

The Optimal Admission Threshold in Observable Queues with State Dependent Pricing. Christian Borgs, Jennifer T. Chayes, Sherwin Doroudi, Mor Harchol-Balter, Kuang Xu. Probability in the Engineering and Informational Sciences, vol. 28, 2014.

Position Paper: Software Techniques for Reducing the Vulnerability of GPU Applications. Si Li, Vilas Sridharan, Sudhanva Gurumurthi, Sudhakar Yalamanchili. Workshop on Dependable GPU Computing (at DATE), March 2014.

Improving Cache Performance Using Read-Write Partitioning. Samira Khan, Alaa Alameldeen, Chris Wilkerson, Onur Mutlu, Daniel Jimenez. Proceedings of the 20th International Symposium on High- Computer Architecture (HPCA'14), February 2014.

MRPB: Memory Request Prioritization for Massively Parallel Processors. Wenhao Jia, Kelly A. Shaw, Margaret Martonosi. Proceedings of the 20th International Symposium on High-Performance Computer Architecture (HPCA'14), February 2014.

Improving DRAM Performance by Parallelizing Refreshes with Accesses. Kevin Chang, Donghyuk Lee, Zeshan Chishti, Chris Wilkerson, Alaa Alameldeen, Yoongu Kim, Onur Mutlu. Proceedings of the 20th International Symposium on High-Performance Computer Architecture (HPCA'14), February 2014.

Toward Strong, Usable Access Control for Shared Distributed Data. Michelle L. Mazurek, Yuan Liang, William Melicher, Manya Sleeper, Lujo Bauer, Gregory R. Ganger, Nitin Gupta, Michael K. Reiter. The 12th USENIX Conference on File and Storage Technologies (FAST'14), February 2014.

Probabilistic Diffusion of Social Influence with Incentives. Myungcheol Doo, Ling Liu. Special Issue on Clouds for Social Computing, IEEE Transactions on Service Computing.

Scalable, High Performance Ethernet Forwarding Lookup. Dong Zhou, Bin Fan, Hyeontaek Lim, Michael Kaminsky, and David G. Andersen, 9th International Conference on emerging Networking Experiments and Technologies (CoNEXT), December 2013. (poster abstract)

Linearly Compressed Pages: A Main Memory Compression Framework with Low Complexity and Low Latency. Gennady Pekhimenko, Vivek Seshadri, Yoongu Kim, Hongyi Xin, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, and Todd C. Mowry, 46th IEEE/ACM International Symposium on Microarchitecture (MICRO-46), December 2013.

Communication-Efficient Distributed Multiple Reference Pattern Matching for M2M Systems. Ruei-Bin Wang, Yu-Chen Lu, Mi-Yen Yeh, Shou-De Lin, and Phillip B. Gibbons, 13th IEEE International Conference on Data Mining (ICDM'13), December 2013.

RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization. Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee, Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, Todd C. Mowry. 46th IEEE/ACM International Symposium on Microarchitecture (MICRO-46), December 2013.

Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation. Yu Cai, Onur Mutlu, Erich F. Haratsch, and Ken Mai, 31st IEEE International Conference on Computer Design (ICCD'13), October 2013.

ISTC-CC Research Overview

continued from pg. 1

and little I/O bandwidth, while others are I/O-bound and involve large amounts of random I/O requests. Some are memory-limited, while others process data in streams (from storage or over the network) with little need for RAM. And, some may have characteristics that can exploit particular hardware assists, such as GPUs, encryption accelerators, and so on. A multi-purpose cloud could easily see a mix of all of these varied application types, and a lowest-common-denominator type configuration will fall far short of best-case efficiency.

We believe that specialization is crucial to achieving the best efficiency—in computer systems, as in any large-scale system (including society), specialization is fundamental to efficiency. Future cloud computing infrastructures will benefit from this concept, purposefully including mixes of different platforms specialized for different classes of applications. Instead of using a single platform configuration to serve all applications, each application (and/or application phase, and/or application component) can be run on available servers that most closely match its particular characteristics. We believe that such an approach can provide order-of-magnitude efficiency gains, where appropriate specialization is applied, while retaining the economies of scale and elastic resource allocation promised by cloud computing.

Additional platforms under consideration include lightweight nodes (such as nodes that use Intel® Atom processors), heterogeneous many-core architectures, and CPUs with integrated graphics, with varied memory, interconnect and storage configurations/technologies. Realizing this vision will require a number of inter-related research activities:

» Understanding important application classes, the trade-offs between them, and formulating specializations to optimize performance.
» Exploring the impact of new platforms based on emerging technologies like non-volatile memory and specialized cores.
» Creating algorithms and frameworks for exploiting such specializations.
» Programming applications so that they are adaptable to different platform characteristics, to maximize the benefits of specialization within clouds regardless of the platforms they offer.

In addition, the heterogeneity inherent to this vision will also require new automation approaches.

Pillar 2: Automation

As computer complexity has grown and system costs have shrunk, operational costs have become a significant factor in the total cost of ownership. Moreover, cloud computing raises the stakes, making the challenges tougher while simultaneously promising benefits that can only be achieved if those challenges are met. Operational costs include human administration, downtime-induced losses, and energy usage. Administration expenses arise from the broad collection of management tasks, including planning and deployment, data protection, problem diagnosis and repair, performance tuning, software upgrades, and so on. Most of these become more difficult with cloud computing, as the scale increases, the workloads run on a given infrastructure become more varied and opaque, workloads mix more (inviting interference), and pre-knowledge of user demands becomes rare rather than expected. And, of course, our introduction of specialization (Pillar 1) aims to take advantage of platforms tailored to particular workloads.

Automation is the key to driving down operational costs. With effective automation, any given IT staff can manage much larger infrastructures. Automation can also reduce losses related to downtime, both by eliminating failures induced by human error (the largest source of failures) and by reducing diagnosis and recovery times, increasing availability. Automation can significantly improve energy efficiency, both by ensuring the right (specialized) platform is used for each application, by improving server utilization, and by actively powering down hardware when it is not needed.

Within this broad pillar, ISTC-CC research will tackle key automation challenges related to efficiency, productivity and robustness, with two primary focus areas:

» Resource scheduling and task placement: devising mechanisms and policies for maximizing several goals including energy efficiency, interference avoidance, and data availability and locality. Such scheduling must accommodate diverse mixes of workloads and frameworks as well as specialized computing platforms.
» Problem diagnosis and mitigation: exploring new techniques for effectively diagnosing and mitigating problems given the anticipated scale and complexity increases coming with future cloud computing.

Pillar 3: Big Data

"Big Data analytics" refers to a rapidly growing style of computing characterized by its reliance on large and often dynamically growing datasets. With massive amounts of data arising from such diverse sources as telescope imagery, medical records, online transaction records, checkout stands and web pages, many researchers and practitioners are discovering that statistical models extracted from data collections promise major advances in science, health care, business efficiencies, and information access. In fact, in domain after domain, statistical approaches are quickly bypassing expertise-based approaches in terms of efficacy and robustness.

The shift toward Big Data analytics pervades large-scale computer usage, from the sciences (e.g., genome sequencing) to business intelligence (e.g., workflow optimization) to data warehousing (e.g., recommendation systems) to medicine (e.g., diagnosis) to Internet services (e.g., social network analysis) and so on. Based on this shift, and their resource demands relative to more traditional activities, we expect Big Data activities to eventually dominate future cloud computing.

We envision future cloud computing infrastructures that efficiently and effectively support Big Data analytics. This requires programming and execution frameworks that provide efficiency

to programmers (in terms of effort to construct and run analytics activities) and the infrastructure (in terms of resources required for given work). In addition to static data corpuses, some analytics will focus partially or entirely on live data feeds (e.g., video or social networks), involving the continuous ingest, integration, and exploitation of new observation data.

ISTC-CC research will devise new frameworks for supporting Big Data analytics in future cloud computing infrastructures. Three particular areas of focus will be:

» "Big Learning" frameworks and systems that more effectively accommodate the advanced machine learning algorithms and interactive processing that will characterize much of next generation Big Data analytics. This includes a focused effort on Big Learning for genome analysis.
» Cloud databases for huge, distributed data corpuses supporting efficient processing and adaptive use of indices. This focus includes supporting datasets that are continuously updated by live feeds, requiring efficient ingest, appropriate consistency models, and use of incremental results.
» Understanding Big Data applications, creating classifications and benchmarks to represent them, and providing support for programmers building them.

Note that these efforts each involve aspects of Automation, and that Big Data applications represent one or more classes for which Specialization is likely warranted. The aspects related to live data feeds, which often originate from client devices and social media applications, lead us into the last pillar.

Pillar 4: To the Edge

Future cloud computing will be a combination of public and private clouds, or hybrid clouds, but will also extend beyond large datacenters that power cloud computing to include billions of clients and edge devices. This includes networking components in select locations and mobile devices closely associated with their users that will be directly involved in many "cloud" activities. These devices will not only use remote cloud resources, as with today's offerings, but they will also contribute to them. Although they offer limited resources of their own, edge devices do serve as bridges to the physical world with sensors, actuators, and "context" that would not otherwise be available. Such physical-world resources and content will be among the most valuable in the cloud.

Effective cloud computing support for edge devices must actively consider location as a first-class and non-fungible property. Location becomes important in several ways. First, sensor data (e.g., video) should be understood in the context of the location (and time, etc.) at which it was captured; this is particularly relevant for applications that seek to pool sensor data from multiple edge devices at a common location. Second, many cloud applications used with edge devices will be interactive in nature, making connectivity and latency critical issues; devices do not always have good connectivity to wide-area networks and communication over long distances increases latency.

We envision future cloud computing infrastructures that adaptively and agilely distribute functionality among core cloud resources (i.e., backend data centers), edge-local cloud resources (e.g., servers in coffee shops, sports arenas, campus buildings, waiting rooms, hotel lobbies, etc.), and edge devices (e.g., mobile handhelds, tablets, netbooks, laptops, and wearables). This requires programming and execution frameworks that allow resource-intensive software components to run in any of these locations, based on location, connectivity, and resource availability. It also requires the ability to rapidly combine information captured at one or more edge devices with other such information and core resources (including data repositories) without losing critical location context.

ISTC-CC research will devise new frameworks for edge/cloud cooperation. Three focus areas will be:

» Enabling and effectively supporting applications whose execution and data span client devices, edge-local cloud resources, and core cloud resources, as discussed above.
» Addressing edge connectivity issues by creating effective data staging and caching techniques that mitigate reliance on expensive and robust Internet uplinks/downlinks for clients, while preserving data consistency requirements.
» Exploring edge architectures, such as resource-poor edge connection points vs. more capable edge-local servers, and platforms for supporting cloud-at-the-edge applications.

Program Director's Corner

Jeff Parkhurst, Intel

It has been a great third year for the Cloud Computing Center. Projects within the center are now getting more attention from Intel. We are seeing good engagement in all 4 themes of the center. We had a great retreat last November and were notified that we were awarded funding for years 4 and 5 and with that, the opportunity to see more of this research mature. With maturing research comes greater interest in engaging from technology stakeholders at Intel. We are always looking to expand this type of engagement and I am happy to facilitate this. If you are an ISTC funded university researcher or an Intel employee looking to engage, please contact me at jeff.parkhurst@intel.com. Here's looking forward to another successful year!

Year in Review

continued from pg. 5

Guardian Has Only Just Begun" at Harbin Institute of Technology, Harbin China, and at Chinese University of Hong Kong, China, May 2014.

» Ion Stoica (UC Berkeley) served as an NSDI'14 Program Co-Chair.

» Onur Mutlu received the Microsoft Research Award for his work on "Improving Datacenter Efficiency and Total Cost of Ownership with Differentiated Software Reliability Analysis and Techniques."

» Karsten Schwan (GA Tech) served as Program Co-chair for the MBDS (Management of Big Data Systems) track of the annual ICAC conference, June 2014.

» Mor Harchol-Balter (CMU) gave the keynote talk entitled "Dynamic Power Management for Data Centers: Theory & Practice" at the GreenMETRICS'14 Workshop, affiliated with the SIGMETRICS Conference, June 2014.

2014 Quarter 3

» Alex Smola (CMU) and his co-author's paper "Reducing the Sampling Complexity of Topic Models" won the best paper award at KDD'14.

» Alex Smola (CMU) co-organized a well-attended two-week summer school on machine learning at CMU.

» Alex Smola's (CMU) presented a tutorial entitled "Scaling Machine Learning" at the Machine Learning Summer School in Pittsburgh, July 2014.

» Guy Blelloch (CMU) gave a series of invited lectures on teaching parallelism at Huazhong University of Science and Technology in Wuhan China, August 2014.

» Alex Smola (CMU), David Andersen (CMU) and group released their parameter server code: http://www.parameterserver.org

» Dave Andersen (CMU) and Michael Kaminsky's (IL) cuckoo hashing code is now part of Intel's internal DPDK repository, with Intel planning to release version 1.8 later this year. Ren Wang (IL) continues to work with Intel SSG to facilitate the integration of the code into DPDK. Dong Zhou, lead CMU grad student on the project, is interning with Ren for the fall.

» The 4th Annual ISTC-CC Retreat will be held in Hillsboro, OR at the Intel Jones Farm campus.

Ren Wang (l), Carlos Rozas (c), and Dan Dahle (r), all of Intel, discuss cloud computing at the 2013 ISTC-CC Retreat.

Message from the PIs

continued from pg. 2

shared ML model parameters). In addition, one of our major capstone efforts will coalesce the knowledge being gained in these activities to lay out a taxonomy of major big-learning styles and the techniques/frameworks that best serve them. Our hope is both to bring some clarity to this still evolving problem space and to ensure that there are no major holes in the solution set, as we move deeper into the data science era.

Another area where major progress is being made is on resource scheduling in clouds, and especially on a topic induced by the cross-section of three ISTC-CC pillars: scheduling for specialization. Our continued efforts to promote platform specialization add a major challenge to the scale and dynamicity of cloud scheduling: workloads are better off when they are assigned to the "right" resources, but they are sometimes happier to get second or third choices rather than waiting for the "right" ones. The challenge is to determine which ones and when, based on quantifying specialization benefits for different workloads. Our new approaches to scheduling are creating the interfaces and automation support needed to make specialization truly effective; pulling together the different aspects of doing this, and combining them with advances, is a big part of our second capstone on resource management for specialization.

While much of ISTC-CC's work focuses on core cloud infrastructure, our Cloudlet project continues to develop technologies for bringing parts of that core closer to the edge. Cool demonstrations focused on cognitive assistance, which is something that demands both significant computing resources and low-latency locality-sensitive turnaround, illustrate the need and serve as strong case studies. (And, Greg continues to wait impatiently for the resulting assistance for his failing wet-ware memory!) We hope that to see this joint effort grow into a third capstone, working together with Intel.

Lots of progress has been made on many other fronts, as well. As one quick example, our Cuckoo hashing software provides fast and memory-efficient key-value storage, and it is being integrated into Intel's DPDK.

As another, our new memory-centric storage architecture greatly improves the efficiency of Big Data analytics without yielding reliability. Our specialization research is yielding new memory system designs and new approaches to robustly exploiting heterogeneity, as well as several summer interns at Intel Labs. And... and... and...

There are too many other examples of cool results, but the news items and paper abstracts throughout this newsletter provide a broader overview. Of course, all of the papers can be found via the ISTC-CC website, and the ISTC-CC researchers are happy to discuss their work. We hope you enjoy the newsletter, and we look forward to sharing ISTC-CC's successes in the months and years to come.