PARALLEL DATA LABORATORY
CARNEGIE MELLON UNIVERSITY

PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • SPRING 2012
http://www.pdl.cmu.edu/

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

CONTENTS

PDL CONSORTIUM MEMBERS

Actifio
American Power Corporation
EMC Corporation
Emulex
Facebook
Fusion-io
Google
Hewlett-Packard Labs
Hitachi, Ltd.
Huawei Technologies Co.
Intel Corporation
Microsoft Research
NEC Laboratories
NetApp, Inc.
Oracle Corporation
Panasas
Riverbed
Samsung Information Systems America
Seagate Technology
STEC, Inc.
Symantec Corporation
VMware, Inc.
Western Digital

Diagnosing Performance Changes by Comparing Request Flows

Raja Sambasivan, Greg Ganger & Joan Digney

Diagnosing performance problems in distributed systems is difficult. Problems may be contained in any one or more component processes or may emerge from the interactions among them. Though there are many tools to help understand the root causes of the diverse types of performance problems that can arise among single-process applications few techniques have been developed for guiding diagnosis of distributed system performance.

Recent research is changing this. Several tools build on low-overhead end-to-end tracing, which captures the flow (i.e., path and timing) of individual requests within and across the components of a distributed system, detecting anomalous request flows or large performance model departures.

Here, we describe a new approach in which performance changes between two executions are identified by comparing their respective request flows. One execution thus serves as a model of acceptable performance, and key differences between the two cases are highlighted. Though obtaining an execution of acceptable performance may not be possible in all cases—e.g., when a developer wants to understand why performance has always been poor—there are many cases for which request-flow comparison is useful. For example, it can help diagnose performance changes resulting from modifications made during software development or from upgrades to components of a deployed system. It can also help when diagnosing changes over time in a deployed system, which may result from component degradations, resource leakage, or workload changes.

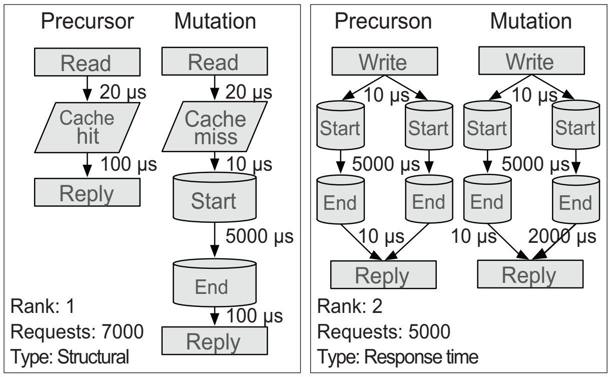


Figure 1. Example output from comparing request flows. The two mutations shown are ranked by their effect on the change in performance. The item ranked first is a structural mutation and the item ranked second is a response-time mutation. Due to space constraints, mocked-up graphs are shown in which nodes represent the type of component accessed.

Request-flow comparison builds on end-to-end tracing, an information source that captures a distributed system's performance and control flow in detail. Such tracing works by capturing activity records at each trace point within the distributed system's software, with each record identifying the specific trace-point name, the current time, and other contextual information. Most implementations

continued on page 17

FROM THE DIRECTOR'S CHAIR

Greg Ganger

Hello from fabulous Pittsburgh!

It has been another great year for the Parallel Data Lab. Some highlights include new projects starting, exciting new results on existing projects, awards for several researchers and papers, and creation of a new cloud computing research center. Along the way, many students graduated and joined PDL Consortium companies, new students have joined PDL, and many papers have been published. Let me highlight a few things.

The most news-grabbing thing from the last year has been the launch of a new research center: the Intel Science and Technology Center for Cloud Computing (ISTC-CC), which focuses on underlying infrastructure for cloud computing. Described more in a short article in this newsletter, ISTC-CC shares a lot of research foci with PDL, such as data-intensive computing (a.k.a. Big Data), automated problem diagnosis, and specialized systems exploiting new storage technologies. Indeed, a lot of PDL faculty play major roles in ISTC-CC, and the two centers amplify one another significantly. We are thrilled to have taken a leadership role in the research needed to realize the promise of cloud computing.

One of today's fastest growing forms of computing now goes by the label "Big Data", and the U.S. government has even made it an explicit research priority. Naturally, PDL has been working on system support for "Big Data" for years, although we've referred to it as "data-intensive computing" or "DISC". Garth's article in this newsletter describes some of our many ongoing activities in this space. Among other things, we continue to operate a DISC service for CMU researchers, based on the Hadoop software stack, and we are exploring ways in which to converge cloud databases and huge-scale parallel file systems—the two are currently evolving separately, but are moving toward similar solutions. Both would benefit, conceptually and practically (e.g., shared software), from creating high-level frameworks and mechanisms that work for both. Our ongoing GIGA+ project, which seeks to support massive directories, offers one such example, using mechanisms much like cloud databases. In continuing work, we are exploring use of such mechanisms both for large-scale metadata services in DISC systems and common high-ingest support for such services and cloud databases generally.

Our field is also seeing rapid and exciting changes in the underlying technologies on which we build. Of course, Flash storage has emerged and is being exploited in many ways (e.g., see the FAWN project), and there is great excitement around forthcoming non-volatile RAM technologies, like PCM and memristors. In addition to exploring interesting uses for them in the storage hierarchy, we are exploring the extent to which substantial fractions of RAM can be replaced with lower-cost-per-bit and more energy-efficient NVRAM technologies. It is also interesting to revisit architectures like "active disks" in the context of Flash-based SSDs, given their internal parallelism and bandwidth characteristics. And, last but not least, even the disk drive is changing, with technologies like shingled magnetic recording creating a need to reconsider usage patterns and interfaces.

We continue to focus a lot of attention on problem diagnosis and use of automation in distributed systems, including DISC systems and large-scale storage. It is clear that there will be no silver bullet here, and PDL research is probing a number of complementary paths. One promising approach involves comparison of request flow graphs, obtained from detailed on-line tracing of work in the

THE PDL PACKET

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

PUBLISHER
Greg Ganger

EDITOR
Joan Digney

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

THE PDL LOGO

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

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

FROM THE DIRECTOR'S CHAIR

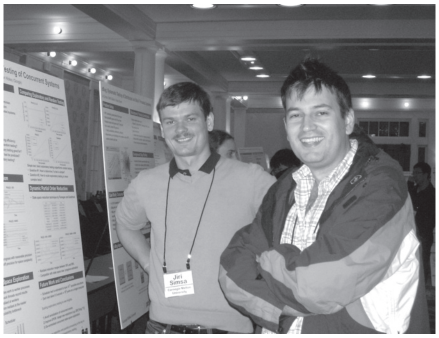
system, across problem and non-problem periods—changes in how given request types are serviced can localize and help explain performance problems in a system. Such tracing is increasingly available in real systems, such as throughout Google's systems. We are also exploring the many other instrumentation sources, such as time-series resource utilization and application logs, and how they can be combined to better deduce where things go wrong. Such data, combined with carefully chosen machine learning algorithms, can decrease the lack of guidance facing humans seeking to diagnose problems.

The FAWN (Fast Array of Wimpy Nodes) project continues to generate exciting results, including yet again winning the 10GB JouleSort benchmark competition in the Daytona and Indy categories. Led by Prof. David Andersen, the FAWN project explores new cluster architectures that can provide data-intensive computing with order of magnitude improvements in energy efficiency. A FAWN cluster uses large collections of embedded processors and Flash memory, rather than smaller collections of high-end servers and disks, providing the same scalability and maximum performance levels while consuming up to one-tenth the power. New prototypes are being built with more specialized components, and research continues to broaden the range of applications that work well on such systems.

Many other ongoing PDL projects are also producing cool results. For example, we have created new data distribution algorithms and policies for allowing elastic sizing of DISC system size, creating the potential for cloud environments efficiently supporting both data analytics and other activities. We continue to operate private clouds in the Data Center Observatory (DCO) for the dual purposes of providing resources for real users (CMU researchers) and providing us with invaluable Hadoop logs, instrumentation data, and case studies. The three clouds—OpenCloud based on Hadoop, OpenCirrus based on Tashi, and vCloud based on VMware's cloud computing software—are helping inform our research into in-cloud data-intensive computing, automated problem diagnosis, and other topics. This newsletter and the PDL website offer more details and additional research highlights.

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

Swapnil Patil presents his work on "YCSB++: Benchmarking and Performance Debugging Advanced Features in Scalable Table Stores" at the 2011 Parallel Data Lab Retreat.



Jiří Šimša and PDL alum, Tudor Dumitraş, (now with Symantec) at a retreat poster session.

FACULTY

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

Anastasia Ailamaki
David Andersen
Lujo Bauer
Chuck Cranor
Lorrie Cranor
Christos Faloutsos
Eugene Fink
Rajeev Gandhi
Garth Gibson
Seth Copen Goldstein
Carlos Guestrin
Mor Harchol-Balter
Bruce Krogh
Todd Mowry
Onur Mutlu
Priya Narasimhan
David O'Hallaron
Adrian Perrig
Mike Reiter
M. Satyanarayanan
Srinivasan Seshan
Bruno Sinopoli
Hui Zhang

STAFF MEMBERS

Bill Courtright, 412•268•5485
(pdl executive director) wcourtright@cmu.edu
Karen Lindenfelser, 412•268•6716
(pdl administrative manager) karen@ece.cmu.edu
Joan Digney
Zisimos Economou
Mitch Franzos
Nitin Gupta
Manish Prasad
Michael Stroucken
Charlene Zang

VISITING RESEARCHER

Norio Shimozono

GRADUATE STUDENTS

Yoshihisa Abe
Rachata Ausavarungnirun
Ben Blum
Jim Cipar
Bin Fan
Bin Fu
Anshul Gandhi
Fan Guo
Junchen Jiang
Wesley Jin
Mike Kasick
Soila Kavulya
Peter Klemperer
Elie Krevat
Hyeontaek Lim
Michelle Mazurek
Justin Meza
Iulian Moraru
Luca Parolini
Swapnil Patil
Amar Phanishayee
Emil Rakadjiev
Kai Ren
Wolfgang Richter
Raja Sambasivan
Vivek Seshadri
Ilari Shafer
Jainam Shah
Jiri Simsa
Anand Suresh
Wittawat Tantisiriroj
Alexey Tumanov
Yifan Wang
Lin Xiao
Lianghong Xu
HanBin Yoon
Xu Zhang
Timothy Zhu

May 2012

- 14th Annual PDL Spring Industry Visit Day.
- Michael Abd-El-Malek's paper, "File System Virtual Appliances: Portable File System Implementations," will appear in ACM Transactions on Storage (TOS), 8.3, any day now.
- Ben Blum is defending his Masters thesis "Landslide: Systematic Dynamic Race Detection in Userspace."
- Ilari Shafer will be interning with Google this summer.
- Yifan Wang presented his Masters thesis "A Statistical Study for File System Metadata On High Performance Computing Sites."

April 2012

- Ilari Shafer and Timothy Zhu were awarded NSF Graduate Research Fellowships.
- Jim Cipar presented "LazyBase: Trading Freshness for Performance in a Scalable Database" at EuroSys 2012 in Bern, Switzerland.

March 2012

- Anshul Gandhi's paper "Minimizing Data Center SLA Violations and Power Consumption via Hybrid Resource Provisioning," has been chosen as Pick of the Month by the IEEE STC on Sustainable Computing Newsletter.
- Garth Gibson received 2012 Jean-Claude Laprie Award in Dependable Computing.
- Michael Kasick proposed "Black-Box Problem Diagnosis in Parallel File Systems" as his thesis topic.
- Soila Pertet Kavulya proposed her thesis research topic "Automated Diagnosis of Chronic Problems in Production Systems."

February 2012

- Garth gave a NetApp CTO's Distinguished Speaker talk on "Recent Work in Storage Systems for BigData."
- Michelle Mazurek presented "ZZFS: A hybrid device and cloud file system for spontaneous users" at FAST 2012 in San Jose, CA.
- Michelle Mazurek and Hyeontaek Lim were awarded Facebook Fellowships.
- Onur Mutlu received the George Tallman Ladd Award for outstanding research.

November 2011

- Wittawat Tantisiriroj presented "On the Duality of Data-intensive File System Design: Reconciling HDFS and PVFS" at Supercomputing 2011 in Seattle, WA.
- 19th Annual Parallel Data Lab Retreat held at Bedford Springs, PA.
- Garth participated in 7th Kavli Futures Symposium, Scalable Energy-Efficient Data Centers and Clouds, Santa Barbara, CA, and presented "Future Needs, Obstacles and Technologies for Storage."

October 2011

- Greg Ganger attended the CERCS Open Cirrus Summit and I/UCRC IAB meeting in Atlanta, GA.
- Vijay Vasudevan defended his Ph.D. dissertation on "Energy-efficient Data-intensive Computing with a Fast Array of Wimpy Nodes."
- Anshul Gandhi presented "The case for sleep states in servers" at HotPower 2011 in Cascais, Portugal.
- Garth Gibson's 1988 RAID Paper, "A Case for Redundant Array of Inexpensive Disks," entered the SIGOPS Hall of Fame.
- U. Kang proposed his thesis research topic "Mining Tera-Scale Graphs with MapReduce: Theory, Engineering and Discoveries."
- Raja Sambasivan proposed his thesis research topic "Diagnosing Performance Changes by Comparing Request Flows."
- Hyeontaek Lim "SILT: A Memory-Efficient, High-Performance Key-Value Store" at SOSP'11 in Cascais, Portugal.
- Swapnil Patil presented "YCSB++: Benchmarking and Performance Debugging Advanced Features in Scalable Table Stores" at the 2nd ACM Symposium on Cloud Computing (SOCC '11) in Cascais, Portugal.
- Bin Fan presented "Small Cache,

continued on page 21

PDL Alums Erik Riedel and Cheryl Gach Riedel and their family, all decked out in the finest PDL fashion.

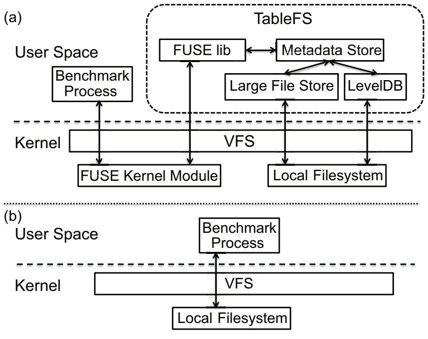
TableFS: Embedding a NoSQL Database Inside the Local File System

Kai Ren & Garth Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-103, April 2012.

While parallel and Internet service file systems have demonstrated effective scaling for high bandwidth, large file transfers in the last decade, the same is not true for workloads that are dominated by metadata and tiny file access. Instead there has emerged a large class of scalable small-data storage systems, commonly called key-value stores, that emphasize simple (NoSQL) interfaces and large in-memory caches.

Some of these key-value stores feature high rates of change and efficient out-of-memory log-structured merge (LSM) tree structures. We assert that file systems should adopt techniques from modern key- value stores for metadata and tiny files, because these systems are "thin" enough to provide performance levels required by file systems. To motivate our assertion, in this paper we present experiments in the most mature and restrictive of environments: a local file system managing one magnetic hard disk. Our results show that for workloads dominated by metadata and tiny files, it is possible to improve the performance of the most modern local file systems in Linux by as much as an order of magnitude by adding an interposed file system layer that represents metadata and tiny files in a LevelDB key-value store that stores its LSM tree and write-ahead log segments in these same local file systems. Perhaps it is finally time to accept the old refrain that file systems should at their core use more database management representations and techniques, now that database management techniques have been sufficiently decoupled from monolithic database management system (DBMS) bundles.



(a) shows the architecture of TABLEFS. A FUSE kernel module redirects file system calls from a benchmark process to TABLEFS, and TABLEFS stores objects into either LevelDB and a large file store. (b) shows the case architecture an experiment compare against in Section 3. These figures suggest the large overhead TABLEFS experiences relative to the traditional local file systems.

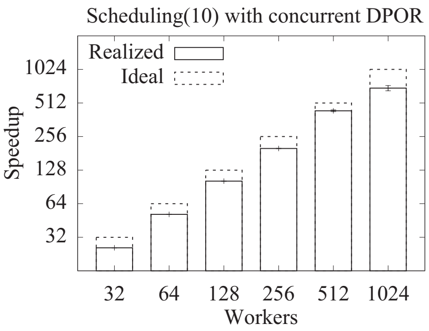
Scalable Dynamic Partial Order Reduction

Jiří Šimša, Randy Bryant, Garth Gibson & Jason Hickey (Google)

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-101. April 2012.

Exploratory testing, first demonstrated in small, specialized cases 15 years ago, has matured sufficiently for large-scale systems developers to begin to put it into practice. With actual deployment comes new, pragmatic challenges to the usefulness of the techniques. In this paper we are concerned with scaling dynamic partial order reduction, a key technique for mitigating the state space explosion problem, to very large clusters.

In particular, we present a new approach for concurrent dynamic partial order reduction. Unlike previous work, our approach is based on a novel exploration algorithm that 1) enables trading space complexity for parallelism, 2) achieves load-balancing through time-slicing, 3) provides for fault tolerance, and 4) has been demonstrated to scale to more than a thousand of concurrent workers.



Scalability measurement for the implementation of our concurrent dynamic partial order reduction. For this example, dynamic partial order reduction explores on the order of 3.6 million different test executions and the sequential implementation is estimated to require 126 hours to accomplish the same task.

LazyBase: Trading Freshness for Performance in a Scalable Database

James Cipar, Greg Ganger, Kimberly Keeton, Charles B. Morrey III, Craig A. N. Soules & Alistair Veitch

EuroSys 2012 April 10-13, 2012, Bern, Switzerland.

The LazyBase scalable database system is specialized for the growing class of data analysis applications that extract knowledge from large, rapidly changing data sets. It provides the scalability of popular NoSQL systems without the query-time complexity associated with their eventual consistency models, offering a clear consistency model and explicit per-query control over the trade-off between latency and result freshness. With an architecture designed around batching and pipelining of updates, LazyBase simultaneously ingests atomic batches of updates at a very high throughput and offers quick read queries to a stale-but-consistent version of the data. Although slightly stale results are sufficient for many analysis queries, fully up-to-date results can be obtained when necessary by also scanning updates still in the pipeline. Compared to the Cassandra

continued on page 6

continued from page 5

NoSQL system, LazyBase provides 4X–5X faster update throughput and 4X faster read query throughput for range queries while remaining competitive for point queries. We demonstrate LazyBase's tradeoff between query latency and result freshness as well as the benefits of its consistency model. We also demonstrate specific cases where Cassandra's consistency model is weaker than LazyBase's.

Active Disk Meets Flash: A Case for Intelligent SSDs

Sangyeun Cho, Chanik Park , Hyunok Oh, Sungchan Kim, Youngmin Yi and Gregory R. Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-11-115. Dec. 2011.

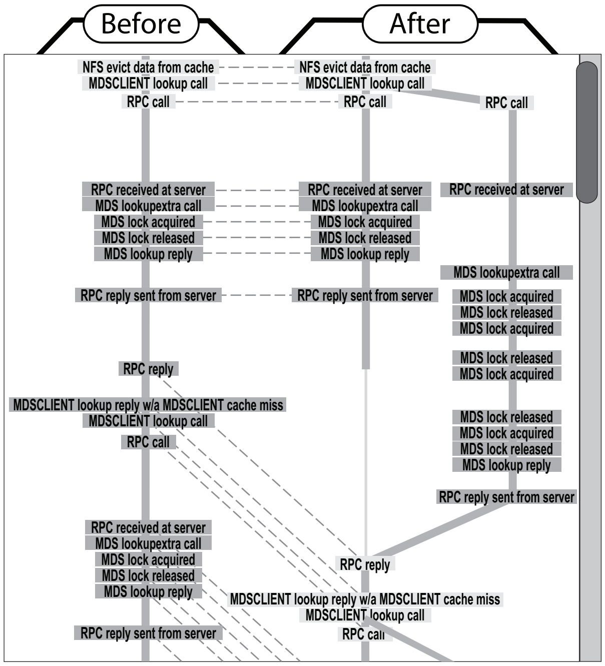
Intelligent solid-state drives (iSSDs) allow execution of limited application functions (e.g., data filtering or aggregation) on their internal hardware resources, exploiting SSD characteristics and trends to provide large and growing performance and energy efficiency benefits. Most notably, internal flash media bandwidth can be significantly (2–4X or more) higher than the external bandwidth with which the SSD is connected to a host system, and the higher internal bandwidth can be exploited within an iSSD. Also, SSD bandwidth is quite high and projected to increase rapidly over time, creating a substantial energy cost for streaming of data to an external CPU for processing, which can be avoided via iSSD processing. This paper makes a case for iSSDs by detailing these trends, quantifying the potential benefits across a range of application activities, describing how SSD architectures could be extended cost-effectively, and demonstrating the concept with measurements of a prototype iSSD running simple data scan functions. Our analyses indicate that, with less than a 4% increase in hardware cost over a traditional SSD, an iSSD can provide 2–4X performance increases and 5–27X energy efficiency gains for a range of data-intensive computations.

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

Raja R. Sambasivan, Ilari Shafer & Gregory R. Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-102, April 2012.

Distributed systems are complex to develop and administer, and performance problem diagnosis is particularly challenging. When performance decreases, the problem might be in any of the system's many components or could be a result of poor interactions among them. Recent research has provided the ability to automatically identify a small set of most likely problem locations, leaving the diagnoser (human) with the task of exploring just that set. This paper describes and evaluates three approaches for visualizing the results of a proven technique called "request-flow comparison" for identifying likely causes of performance decreases in a distributed system. Our user study provides a number of insights useful in guiding visualization tool design for distributed system diagnosis. For example, we find that both an overlay-based approach (e.g., diff) and a side-by-side approach are effective, with tradeoffs for different users (e.g., expert vs. not) and different problem types. We also find that an animation-based approach is confusing and difficult to use.



Comparing request-flow graphs: This side-by-side visualization is one of the three interfaces evaluated for showing the output of request-flow comparison, a performance diagnosis technique that identifies performance-affecting differences in the flow of requests within and among the components of a distributed system. The interface shows two graphs (labeled "before" and "after") juxtaposed horizontally, with dashed lines indicating nodes that are the same between them. The rightmost series of nodes in the after graph do not exist in the before graph, causing the yellow nodes to shift downward in the after graph.

Persistent, Protected and Cached: Building Blocks for Main Memory Data Stores

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

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-11-114. Dec. 2011.

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,

continued on page 7

continued from page 6

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 today's storage technologies. This paper introduces techniques for (1) robust wear-aware memory allocation, (2) prevention 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.

Towards Understanding Heterogeneous Clouds At Scale: Google Trace Analysis

Charles Reiss (UCBerkeley), Alexey Tumanov, Gregory R. Ganger, Randy H. Katz (UCBerkeley), Micheal A. Kozuch (Intel Labs)

Intel Science and Technology Center for Cloud Computing Technical Report ISTC-CC-TR-12-101, April 2012.

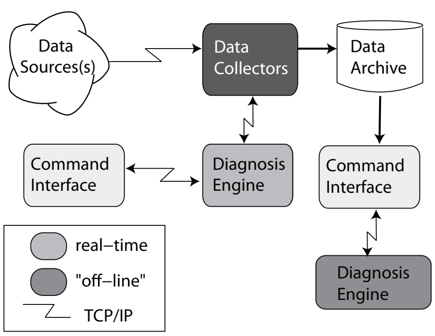
With the emergence of large, *heterogeneous*, shared computing clusters, their efficient use by *mixed* distributed workloads and tenants remains an important challenge. Until now, very little data was available about workloads in production clusters of such scale and with such detail and resolution. This paper analyzes a recent Google release of scheduler request and utilization data across a large (12500+) general-purpose compute cluster over 29 days. We characterize cluster resource requests, their distribution, and the actual resource utilization. Unlike previous scheduler traces we are aware of, this one includes diverse workloads—from large web services to large CPU-intensive batch programs—and permits comparison of actual resource utilization with the user-supplied resource estimates available to the cluster resource scheduler. We observe consistent underutilization despite overcommitment of resources, difficulty of scheduling high-priority tasks when they are constrained, and lack of dynamic adjustments to user allocation requests despite the apparent availability of this feature in the scheduler.

Draco: Statistical Diagnosis of Chronic Problems in Large Distributed Systems.

Soila P. Kavulya, Scott Daniels (AT&T), Kautubh Joshi (AT&T), Matti Hiltunen (AT&T), Rajeev Gandhi & Priya Narasimhan

IEEE/IFIP Conference on Dependable Systems and Networks (DSN), June 2012.

Chronics are recurrent problems that often fly under the radar of operations teams because they do not affect enough users or service invocations to set off alarm thresholds. In contrast with major outages that are rare, often have a single cause, and as a result are relatively easy to detect and diagnose quickly, chronic problems are elusive because they are often triggered by complex conditions, persist in a system for days or weeks, and coexist with other problems active at the same time. In this paper, we present Draco, a scalable engine to diagnose chronics that addresses these issues by using a "topdown" approach that starts by heuristically identifying user interactions that are likely to have failed, e.g., dropped calls, and drills down to identify groups of properties that best explain the difference between failed and successful interactions by using a scalable Bayesian learner. We have deployed Draco in production for the VoIP operations of a major ISP. In addition to providing examples of chronics that Draco has helped identify, we show via a comprehensive evaluation on production data that Draco provided 97% coverage, had fewer than 4% false positives, and outperformed state-of-the-art diagnostic techniques by up to 56% for complex chronics.



Draco's flexible architecture supports multiple data sources, and the diagnosis engines can run in either real-time or offline mode.

ZZFS: A Hybrid Device and Cloud File System for Spontaneous Users

Michelle L. Mazurek, Eno Thereska, Dinan Gundawardena, Richard Harper & James Scott

FAST 2012: USENIX Conference on File and Storage Technologies, February 2012.

Good execution of data placement, caching and consistency policies across a user's personal devices has always been hard. Unpredictable networks, capricious user behavior with leaving devices on or off and non-uniform energy-saving policies constantly interfere with the good intentions of a storage system's policies. This paper's contribution is to better manage these inherent uncertainties. We do so primarily by building a low-power communication channel that is available even when a device is off. This channel is mainly made possible by a novel network interface card that is carefully placed under the control of storage system protocols. The design space can benefit existing placement policies (e.g., Cimbiosys [21], Perspective [23], Anzere [22]). It also allows for interesting new ones. We build a file system called ZZFS around a particular set of policies motivated by user studies. Its policies cater to users who interact with the file system in an ad hoc way — spontaneously and without pre-planning.

continued on page 22

April 2012

Ilari Shafer and Timothy Zhu Awarded NSF Graduate Fellowships

Congratulations to Ilari and Timmy, who have been awarded prestigious NSF Graduate Research Fellowships for 2012. The NSF Graduate Research Fellowship Program (GRFP) helps ensure the vitality of the human resource base of science and engineering in the United States and reinforces its diversity. The program recognizes and supports outstanding graduate students in NSF-supported science, technology, engineering, and mathematics disciplines who are pursuing research-based master's and doctoral degrees at accredited U.S. institutions.

Fellows share in the prestige and opportunities that become available when they are selected. Fellows benefit from a three-year annual stipend of $30,000 along with a $10,500 cost of education allowance for tuition and fees, opportunities for international research and professional development.

-- info from http://nsfgrfp.org

March 2012

Anshul Gandhi wins Best Paper

Anshul Gandhi's paper "Minimizing Data Center SLA Violations and Power Consumption via Hybrid Resource Provisioning," which won the Best Paper Award at the 2nd International Green Computing Conference (IGCC 2011) in Orlando, FL last July, was also selected as the Pick of the Month for March 2012 in the IEEE STC on Sustainable Computing newsletter.

March 2012

Garth Gibson Receives 2012 Jean-Claude Laprie Award in Dependable Computing

Garth has been awarded the 2012 Jean-Claude Laprie Award in Dependable Computing, Industrial/Commercial Product Impact Category, by the IFIP Working Group 10.4 on Dependable Computing and Fault Tolerance. The award is for outstanding papers published at least 10 years ago that have significantly influenced the theory and/or practice of dependable computing, and given for "A Case for Redundant Arrays of Inexpensive Disks (RAID)," by D.A. Patterson, G.A. Gibson, and R.H. Katz, Proc. of 1988 ACM SIGMOD Int. Conf. on Management of Data, June 2, 1998. The groundbreaking paper introduced the concept of RAID, which rapidly became the common configuration paradigm for disks at all but the very low end of the server market. Its impact is primarily to industry where RAID was a truly disruptive technology. The RAID levels as defined in this paper persist to the present day. The paper familiarized development engineers who didn't normally work in the area of High Availability or Fault Tolerance with the concepts of improving reliability and availability through redundancy.

The award will be made at the 42nd Annual IEEE/IFIP Dependable Systems and Networks Conference (DSN), Boston, MA, June 25-28, 2012.

-- info from http://www.dependability.org/articles/laprie/laprie2012.html

February 2012

Onur Mutlu receives George Tallman Ladd Award

Onur Mutlu, assistant professor of electrical and computer engineering has received the George Tallman Ladd Research Award. The G.T. Ladd award is made to a faculty member within the Carnegie Institute of Technology in recognition of outstanding research and professional accomplishments and potential. The award is in the form of a memento and an honorarium. The award is made each year and is made based on excellence in research as measured by scholarly publications, research program development, development of funding, and awards and other recognition. Congratulations Onur!

February 2012

Michelle Mazurek and Hyeontaek Lim Facebook Fellowship Winners!

From among 300 applications, two PDL students have been named winners of a 2012-2013 Facebook Fellowship (12 awarded). Hyeontaek is working to improve the resource efficiency of distributed systems. He hopes to deliver more affordable data-intensive computing, facilitating future innovations for large-scale Internet services. Michelle is researching ways to let users share their content accurately and quickly, secure in the knowledge that only the right people will see it.

The fellowship program began in 2010 to "foster ties to the academic community and support the research of promising computer science Ph.D. students." Each student will be granted full tuition, a $30,000 stipend, $5000 for conference travel and $2500 for a new computer.

One other PDL student, Bin Fan,

continued on page 9

continued from page 8

was named one of 30 finalists. Great work all!

-- with info from http://on.fb.me/wjiCZZ

January 2012

Congratulations Swapnil and Shilpa!

Swapnil Patil and Shilpa Deshmukh were wed at the Corinthians Club in Pune, India on January 17, 2012. PDL was well-represented at the wedding—Jiri Simsa, Vivek Seshadri and Milo Polte made the long trip to India to enjoy traditional Indian wedding food and festivities.

They are currently enjoying their last few months in Pittsburgh before they move to the Bay Area in Fall 2012—Swapnil has accepted an offer to join Google (systems infrastructure team) in Mountain View, CA!

October 2011

Garth's 1988 RAID Paper Enters SIGOPS Hall of Fame

We are very pleased to announce that Garth Gibson's original RAID paper from SIGMOD 1988—"A Case for Redundant Array of Inexpensive Disks" by Patterson, Gibson and Katz—was one of the four papers to be honored as a 2011 SIGOPS Hall of Fame Award paper. The award was made at the 23rd ACM Symposium on Operating Systems Principles (SOSP), October 23-26, 2011, Cascais, Portugal.

The SIGOPS Hall of Fame Award was instituted in 2005 to recognize the most influential Operating Systems papers that were published at least ten years in the past. The Hall of Fame Award Committee consists of past program chairs from SOSP, OSDI, EuroSys, past Weiser and Turing Award winners from the SIGOPS community, and representatives of each of the Hall of Fame Award papers.

October 2011

Welcome Christina!

Eno Thereska and his wife Eszter Linczmayer are thrilled to welcome their daughter Christina Lotti Thereska. She was born on October 10, bright and early in the morning and weighed 7.4 lbs. Eno says she keeps them busy and very tired.

August 2011

Intel Labs Invests $30M in the Future of Cloud and Embedded Computing with the Opening of Latest Intel Science and Technology Centers

SANTA CLARA, CA, August 3, 2011–Aimed at shaping the future of cloud computing and how increasing numbers of everyday devices will add computing capabilities, Intel Labs announced the latest Intel Science and Technology Centers (ISTC) for Cloud Computing Research and for Embedded Computing, both headquartered at Carnegie Mellon University.

Intel Science & Technology Center for Cloud Computing

The ISTC for Cloud Computing forms a new cloud computing research community that broadens Intel's "Cloud 2015" vision with new ideas from top academic researchers, and includes research that extends and improves on Intel's existing cloud computing initiatives. The center combines top researchers from Carnegie Mellon University, Georgia Institute of Technology, University of California Berkeley, Princeton University, and Intel. The researchers will explore technology that will have has important future implications for the cloud, including built-in application optimization, more efficient and effective support of big data analytics on massive amounts of online data, and making the cloud more distributed and localized by extending cloud capabilities to the network edge and even to client devices.

In the future, these capabilities could enable a digital personal handler via a device wired into your glasses that sees what you see, to constantly pull data from the cloud and whisper information to you during the day—telling you who people are, where to buy an item you just saw, or how to adjust your plans when something new comes up.

-- from Intel News Room, by C. Brown

August 2011

Welcome Atharv!

Atharv Krish Gupta was born to Nitin and Sumedha Gupta on August 21 at 9:16 PM. He weighed 6 lb. 8 oz. and was 20.5 inches tall at birth. His interests include books, computers, smartphones and electrical cables.

continued on page 10

continued from page 9

June 2011

Satya Receives Outstanding Contributions Award at Mobisys'11

Congratulations to Prof. M. Satyanarayanan (Satya), who was awarded the SIGMOBILE 2010 Outstanding Contributions Award "for pioneering a wide spectrum of technologies in support of disconnected and weakly connected mobile clients" at Mobisys 2011. He joins an illustrious group of previous winners, including Prof. Daniel P. Siewiorek, who received the award in 2006 "for pioneering fundamental contributions to wearable and context-aware computing." The SIGMOBILE Outstanding Contribution Award is given for significant and lasting contributions to the research on mobile computing and communications, and wireless networking.

June 2011

Onur Mutlu wins IEEE Young Computer Architect Award

ECE Assistant Professor Onur Mutlu has earned the inaugural IEEE Computer Society Technical Committee on Computer Architecture's Young Computer Architect Award "in recognition of outstanding contributions in the field of computer architecture in both research and education." The award recognizes outstanding contributions in the field of computer architecture by an individual who received their Ph.D. within six years of their nomination.

-- 8.5x11 News, June 23, 2011

June 2011

PDL Alums win Best Demonstration at SIGMOD 2011

The demonstration of the DORA system ("A Data-oriented Transaction Execution Engine and Supporting Tools") won the Best Demonstration Award at SIGMOD 2011! The team that implemented the demo consisted of Ippokratis Pandis, Pinar Tozun, Miguel Branco, Dimitris Karampinas, Danica Porobic, Ryan Johnson and Anastasia Ailamaki. The entire team is now affiliated with EPFL, with Ippokratis, Ryan and Anastasia all recent members of the PDL. SIGMOD is the premier conference on data management systems, this year held in Athens, Greece.

June 2011

Swapnil Patil Receives ACM Student Research Award!

Swapnil Patil, a PhD student (CSD), took first place in the graduate student category of the Association for Computing Machinery (ACM) Student Research Competition Grand Finals. Patil received the award June 4 at the ACM Awards Banquet in San Jose, Calif., for his development of a file system director service that scales to millions of files, which he presented at SC10, the international conference for high performance computing, networking, storage and analysis. ACM's Student Research Program is sponsored by Microsoft Research to encourage students to pursue careers in computer science research, and to ensure the future of scientific discovery and innovation.

June 2011

FAWN Team Wins of 2011 10GB JouleSort Daytona and Indy Categories

The FAWN team, a joint Intel-CMU group, including Padmanabhan Pillai, Michael Kaminsky, Michael A. Kozuch, Vijay Vasudevan, Lawrence Tan and David G. Andersen won the 2011 10GB JouleSort competition using a Sandy Bridge-based platform with Intel SSDs. For more details see FAWNSort: Energy-efficient Sorting of 10GB and the Sort Benchmark Home Page.

A BRIEF INTRODUCTION TO THE INTEL SCIENCE & TECHNOLOGY CENTER FOR CLOUD COMPUTING (ISTC-CC)

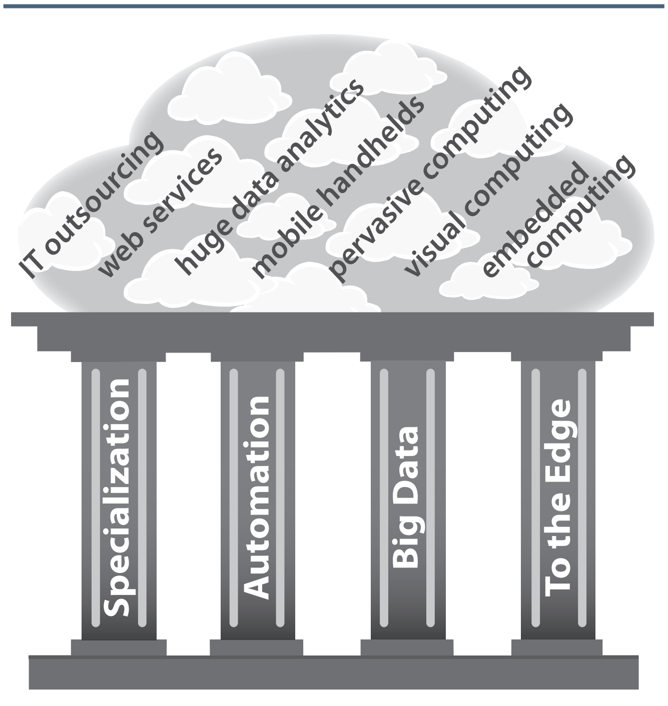
Greg Ganger

Cloud computing has become a source of enormous buzz and attention in the industry, promising great reductions in the effort of establishing new applications/services, increases in the efficiency of operating them, and improvements in the ability to share data and services. Despite the hype and energy around it, though, cloud computing is in its nascent stage—little has been demonstrated, and much is yet to be figured out. Although the idea of computing as a utility has been around for almost half a century, only recently have there been successful offerings at substantial scale. Companies are creating cloud building blocks and services, but are making things up as they go and changing regularly as experience is gained. Much is yet to be learned.

Realizing the promise of cloud computing will require an enormous amount of research and development across a broad array of topics. ISTC-CC was launched in August 2011 to address a critical part of this need: underlying cloud infrastructure technologies to serve as a robust, efficient foundation for cloud applications. Briefly stated, ISTC-CC's mission is to explore system architectures,

continued on page 11

continued from page 10



Four inter-related research pillars provide a strong foundation for cloud computing of the future.

programming models, automation mechanisms, and related technologies that enable dramatic efficiency, ubiquity, and productivity improvements in cloud computing.

ISTC-CC is an open community of leading researchers devising and studying critical new underlying technologies for future clouds and cloud applications. With over $15M of Intel investment and joint leadership by Greg Ganger (Carnegie Mellon) and Phil Gibbons (Intel Labs), ISTC-CC is headquartered at Carnegie Mellon University and includes a broad community of openly collaborating researchers from Carnegie Mellon, Georgia Tech, Intel, Princeton, UC-Berkeley, and many other institutions. Intel has also created several other ISTCs focused on other topics, such as pervasive computing (headquartered at Univerisity of Washington) and visual computing (headquartered at Stanford).

The 5-year 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:

Specialization. Contrary to the common practice of striving for homogeneous cloud deployments, clouds should embrace heterogeneity, purposely including mixes of different platforms specialized for different classes of applications. This pillar explores the use of specialization as a primary means for order of magnitude improvements in efficiency (e.g., energy), including new platform designs based on emerging technologies like non-volatile memory and specialized cores.

Automation. Automation is key to driving down the operational costs (human administration, downtime-induced losses, and energy usage) of cloud computing. The scale, diversity, and unpredictability of cloud workloads increase both the need for, and the challenge of, automation. This pillar addresses cloud's particular automation challenges, focusing on order of magnitude efficiency gains from smart resource allocation/scheduling (including automated selection among specialized platforms) and greatly improved problem diagnosis capabilities.

Big Data. Cloud activities of the future will be dominated by analytics over large and growing data corpuses. This pillar 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 twitter).

To the Edge. Future cloud computing will extend beyond centralized (back-end) resources by encompassing billions of clients and edge devices. The sensors, actuators, and "context" provided by such devices will be among the most valuable content/resources in the cloud. This pillar explores new frameworks for edge/cloud cooperation that (i) can efficiently and effectively exploit this "physical world" content in the cloud, and (ii) enable cloud-assisted client computations, i.e., applications whose execution spans client devices, edge-local cloud resources, and core cloud resources.

Clearly, there is substantial overlap between the ISTC-CC research agenda and PDL research activities. This is a good thing, and I think of it like a Venn diagram. Some Carnegie Mellon research activities are part of PDL and not ISTC-CC, some are ISTC-CC and not PDL, and some are both. For those that are both, PDL and ISTC-CC benefit directly from amplification as their funding combines to allow broader and deeper explorations. Each also benefits, indirectly, from the non-overlapping activities of the other. We look forward to seeing and sharing great research from these activities, in the coming years.

Raja Sambasivan discusses his work on "Diagnosing Performance Changes By Comparing Request Flows."

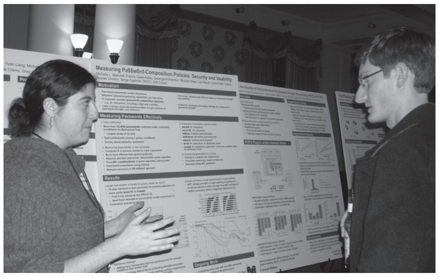
DISSERTATION ABSTRACT: Energy-efficient Data-intensive Computing with a Fast Array of Wimpy Nodes

Vijay Vasudevan

Carnegie Mellon University SCS
Ph.D. Dissertation, Oct. 10, 2011

Large-scale data-intensive computing systems have become a critical foundation for Internet-scale services. Their widespread growth during the past decade has raised datacenter energy demand and created an increasingly large financial burden and scaling challenge: Peak energy requirements today are a significant cost of provisioning and operating datacenters. In this thesis, we propose to reduce the peak energy consumption of datacenters by using a FAWN: A Fast Array of Wimpy Nodes. FAWN is an approach to building datacenter server clusters using low-cost, low-power servers that are individually optimized for energy efficiency rather than raw performance alone. FAWN systems, however, have a different set of resource constraints than traditional systems that can prevent existing software from reaping the improved energy efficiency benefits FAWN systems can provide.

This dissertation describes the principles behind FAWN and the software techniques necessary to unlock its energy efficiency potential. First, we present a deep study into building FAWN-KV, a distributed, log-structured key-value storage system designed for an early FAWN prototype. Second, we present a broader classification and workload analysis showing when FAWN can be more energy-efficient and under what workload conditions a FAWN cluster would perform poorly in comparison to a smaller number of high-speed systems. Last, we describe modern trends that portend a narrowing gap between CPU and I/O capability and highlight the challenges endemic to all future balanced systems. Using FAWN as an early example, we demonstrate that pervasive use of "vector interfaces" throughout distributed storage systems can improve throughput by an order of magnitude and eliminate the redundant work found in many data-intensive workloads.



Michelle Mazurek discusses her poster "Tag, You Can See It! Using Tags for Access Control in Photo Sharing" with Craig Soules (HP Labs) at the 2011 PDL Retreat.

DISSERTATION ABSTRACT: Mining and Querying Multimedia Data

Fan Guo

Carnegie Mellon University SCS
Ph.D. Dissertation, Sept. 19, 2011

The emerging popularity of multimedia data, as digital representation of text, image, video and countless other milieus, with prodigious volumes and wild diversity, exhibits the phenomenal impact of modern technologies in reforming the way information is accessed, disseminated, digested and retained. This has iteratively ignited the data-driven perspective of research and development, to characterize perspicuous patterns, crystallize informative insights, and realize elevated experience for end-users, where innovations in a spectrum of areas of computer science, including databases, distributed systems, machine learning, vision, speech and natural languages, has been incessantly absorbed and integrated to elicit the extent and efficacy of contemporary and future multimedia applications and solutions.

Under the theme of pattern mining and similarity querying, this manuscript presents a number of pieces of research concerning multimedia data, to address an array of practical tasks encompassing automatic annotation, outlier detection, community discovery, multi-modal retrieval and learning to rank, in their respective contexts including satellite image analysis, internet traffic surveillance, image bioinformatics, and Web search. A repertoire of extant and novel techniques pertaining to graph mining, clustering analysis, tensor decomposition and probabilistic graphical models has been developed or adapted, which satisfactorily met differing quality and efficiency requisites postulated by specific application settings, best exemplified by the 40 times speed-up in annotating satellite images and the up to 30% performance improvement in predicting web search user clicks, yet without the loss of generality to similar and related scenarios.

DISSERTATION ABSTRACT: Performance Insulation: More Predictable Shared Storage

Matthew Wachs

Carnegie Mellon University SCS
Ph.D. Dissertation, Sept. 28, 2011

Many storage workloads do not need the performance afforded by a dedicated storage system, but do need the predictability and controllability that comes from one. Unfortunately, inter-workload interference, such as a reduction of locality when multiple request streams are interleaved, can result in dramatic loss of efficiency and performance.

Performance insulation is a system property where each workload sharing the system is assigned a fraction of resources (such as disk time) and receives nearly that fraction of its standalone (dedicated system) performance. Be-

continued on page 13

continued from page 12

cause there is usually some overhead caused by sharing, there could be a drop in efficiency; but a system providing performance insulation provides a bound on efficiency loss at all times, called the R-value. We have built a storage server called Argon that achieves performance insulation in practice for R-values of 0.8-0.9. This means that, running together with other workloads on Argon, workloads lose, at most, only 10-20% of the efficiency they receive on a dedicated system.

While performance insulation provides a useful limit on loss of efficiency, many storage workloads also need performance guarantees. To ensure performance guarantees are consistently met, the appropriate allocation of resources needs to be determined and reserved, and later reevaluated if the workload changes in behavior or if the interference between workloads affects their ability to use resources effectively. If the resources assigned to a workload need to be increased to maintain its guarantee, but adequate resources are not available, violations will result.

Though intrinsic workload variability is fundamental, storage systems with the property of performance insulation strictly limit inter-workload interference, another source of variability. Such interference is the major source of "artificial" complexity in maintaining performance guarantees. We design and evaluate a storage system called Cesium that limits interference and thus avoids the class of guarantee violations arising from it. Workloads running on Cesium only suffer from those violations caused by their own variability and not those due to the activities of other workloads. Realistic and challenging workloads may experience an order of magnitude fewer violations running under Cesium. Performance insulation thus results in more reliable and efficient bandwidth guarantees.

DISSERTATION ABSTRACT: Fast Algorithms for Mining Co-evolving Time Series

Lei Li

Carnegie Mellon University SCS
Ph.D. Dissertation, Sept. 17, 2011

Time series data arise in numerous applications, such as motion capture, computer network monitoring, data center monitoring, environmental monitoring and many more. Finding patterns and learning features in such collections of sequences are crucial to solve real-world, domain specific problems, for example, to build humanoid robots, to detect pollution in drinking water, and to identify intrusion in computer networks.

In this thesis, we focus on fast algorithms on mining co-evolving time series, with or without missing values. We will present a series of our effort in analyzing those data: (a) time series mining and summarization with missing values, and (b) learning features from multiple sequences. Algorithms proposed in the first work allow us to obtain meaningful patterns effectively and efficiently. Thus they enable vital mining tasks including forecast, compression, and segmentation for co-evolving time series, even with missing values. We also propose "PLiF" and Complex Linear Dynamical System (CLDS), novel algorithms to extract features from multiple sequences. Such features will serve as a corner stone of many applications for time series such clustering and similarity search. Our algorithms scale linearly with respect to the length of sequences, and outperform the competitors often by large factors. In addition, we will briefly mention several other time series mining problems and algorithms, including natural motion stitching, bone constrained occlusion filling, a parallelization of our algorithms for multi-core systems, and an forecasting algorithm for thermal conditions in data centers.

Bill Courtright and Ed Gronke (Panasas) discuss storage systems at the retreat.

DISSERTATION ABSTRACT: Scalable Transaction Processing through Data-oriented Execution

Ippokratis Pandis

Carnegie Mellon University SCS
Ph.D. Dissertation, May 12, 2011

Data management technology changes the world we live in by providing efficient access to huge volumes of constantly changing data and by enabling sophisticated analysis of those data. While there has been an unprecedented increase in the demand for data management services; in parallel, we witness a tremendous shift in the underlying hardware toward highly parallel multicore processors. The data management systems in order to cope with the increased demand and user expectations, they need to exploit fully the abundantly available hardware parallelism. Transaction processing is one of the most important and challenging database workloads and this dissertation contributes in the quest for scalable transaction processing software. It shows that in the highly parallel multicore landscape the system designers should primarily focus on reducing the un-scalable critical sections of their systems, rather than improving the single-thread performance. In addition, it makes solid improvements in conventional transaction processing technology by avoiding executing unscalable critical sections in the lock manager through caching, and in the log manager by downgrading them to

continued on page 14

continued from page 13

composable ones. More importantly, it shows that conventional transaction processing has inherent scalability limitations due to the unpredictable access patterns caused by the request-oriented execution model it follows. Instead, it proposes to adopt a data-oriented execution model, and shows that transaction processing systems designed around data-oriented transaction execution break the inherent limitations of conventional execution. The data-oriented design paves the way for transaction processing systems to maintain scalability as parallelism increases for the foreseeable future; as hardware parallelism increases the benefits will only increase. In addition, the principles used to achieve scalability can generalize to other software systems facing similar scalability challenges with the shift to multicore hardware.

THESIS PROPOSAL: Black-Box Problem Diagnosis in Parallel File Systems

Michael Kasick, ECE

March 29, 2012

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

As a solution I propose a problem diagnosis approach, capitalizing upon the parallel file system design criterion of balanced performance, that peer-compares the performance of cluster components and automatically diagnoses problems within storage clusters running unmodified, "off-the-shelf" parallel file systems. Specifically this approach, implementable as a tool, seeks to (i) detect the existence of problems, (ii) localize problems to specific storage cluster components, (iii) determine which problems to report as most severe. Through these steps, this diagnosis approach seeks to aid operators in diagnosing, triaging, and remedying problems that occur within their storage clusters.

THESIS PROPOSAL: Automated Diagnosis of Chronic Problems in Production Systems

Soila Pertet Kavulya, ECE

March 27, 2012

Large distributed systems are susceptible to chronic problems—performance degradations or exceptions which occur intermittently or affect a small subset of end users. Chronics are elusive to diagnose because these problems often fly under the radar of operations teams as they may not be severe enough to set off alarm thresholds. Operators could ignore these problems if they were one-off incidents. However, the recurrent nature of these problems negatively impacts user satisfaction over time. I propose a "top-down" approach to diagnosing chronics in distributed systems that starts by identifying user-visible symptoms of a problem, and drilling down to identify the components and associated resource-usage metrics (e.g., CPU, memory) that are the most highly indicative of the symptoms. My approach is well-suited for production environments as it uses unmodified application-level and system-level logs for diagnosis. I validate my approach using fault-injection experiments and analysis of real incidents in two production systems namely: the Hadoop parallel-processing framework, and a production Voice-over-IP system at a large telecommunications provider.

PDL graduate students Bin Fu, Xu Zhang and Lianghong Xu at the 2011 PDL Rereat.

THESIS PROPOSAL: Mining Tera-Scale Graphs with MapReduce: Theory, Engineering and Discoveries

U. Kang, SCS

October 20, 2011

How do we find patterns and anomalies, on graphs with billions of nodes and edges, which do not fit in memory? How to use parallelism for such Tera- or Peta-scale graphs? In this thesis, we propose a carefully selected set of fundamental operations, that help answer those questions, including diameter estimation, solving eigenvalues, and inference on graphs. We package all these operations in PEGASUS, which, to the best of our knowledge, is the first such library, implemented on the top of the HADOOP platform, the open source version of MAPREDUCE. One of the key observations in this thesis is that many graph mining operations are essentially repeated matrix-vector multiplications. We describe a very important primitive for PEGASUS, called GIM-V (Generalized Iterated Matrix-Vector multiplication). GIM-V is highly optimized, achieving (a) good scale-up on the number of available

continued on page 15

continued from page 14

machines, (b) linear running time on the number of edges, and (c) more than 9 times faster performance over the non-optimized version of GIM-V. Finally, we run experiments on real graphs. Our experiments ran on DiscCloud and M45, one of the largest HADOOP clusters available to academia. We report our findings on several real graphs, including one of the largest publicly available Web graphs, thanks to Yahoo! with ~6,7 billion edges. Some of our most impressive findings are (a) the discovery of adult advertisers in the who-follows-whom on Twitter, and (b) the 7-degrees of separation in the Web graph. Based on our current work, we propose the followings: large scale tensor analysis, graph layout for better compression, and anomaly detection in network data.

THESIS PROPOSAL: Diagnosing Performance Changes by Comparing Request Flows

Raja Sambasivan, SCS

October 14, 2011

The causes of performance changes in a distributed system often elude even its developers. This proposed thesis develops a new technique for gaining insight into such changes: comparing request flows from two executions (e.g., of two system versions or time periods). Building on end-to-end request flow tracing within and across components, algorithms are described for identifying and ranking changes in the flow and/or timing of request processing. The implementation of these algorithms in a tool called Spectroscope is described and evaluated. Eight case studies are presented of using Spectroscope to diagnose performance changes in a prototype distributed storage service and in select Google services. To further show the generality of request-flow comparison, we also propose to adapt Spectroscope to work with HDFS and diagnose real problems observed within it.

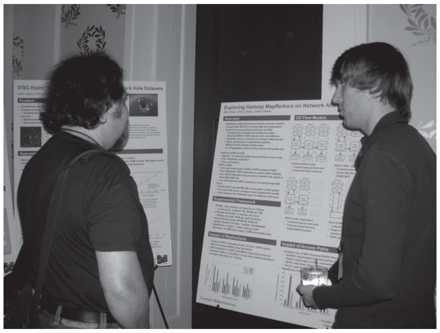
THESIS PROPOSAL: One Key-value System to Store Them All!

Amar Phanishayee, SCS

June 20, 2011

To meet their needs for cost-effective high performance data access and analytics, many large sites such as Amazon, LiveJournal, Facebook, and Twitter turn to simpler data model "NoSQL" systems. Unfortunately, even within one popular sub-category of these solutions—key-value (KV) storage systems—no one system meets the needs of all applications. Application requirements sit on a multi-dimensional continuum, with the breadth of "NoSQL" systems testifying to the value of finding a design and implementation well matched to one's requirements. We argue that having systems designers worry about running multiple stores from different codebases, vendors, configurations, and so on, each optimized for certain application requirements and hardware configuration, is unreasonable and unnecessary. This thesis proposes that it possible for a single key-value system to flexibly meet the needs of a variety of applications running on different hardware configurations, and to be easily configured to support many points along the continuum, from weakly-consistent, non-replicated caches to strongly-consistent, durable disk-backed key-value stores.

This proposal focuses on the following research questions: First, we address the question of designing a cluster key-value store that achieves strong consistency and high performance across a wide range of hardware choices and cluster sizes. We present FAWN-KV which offers replication and strong consistency by using a novel variant of chain replication on a consistent hashing ring, designed to minimize blocking during node additions and deletions. We use FAWN as a motivating hardware configuration to evaluate FAWN-KV.


Ellis Wilson explains his research on "Exploring the Use of Hadoop MapReduce on Network Attached Storage" to Ed Gronke (Panasas).

Second, we address whether it possible for a single key-value system to flexibly meet the needs of a variety of applications. In this thesis, we make the case for a flexible key-value storage system that can support both DRAM and disk-based storage, can act as an unreliable cache or a durable store, and operate consistently or inconsistently. The value of such a system goes beyond ease-of-use: While exploring these dimensions of durability, consistency, and availability, we find new choices for system designs, such as a cache-consistent memcached, that offer some applications a better balance of performance and cost than was previously available.

THESIS PROPOSAL: Systematic and Scalable Testing of Concurrent Systems

Jiří Šimša, SCS

April 28, 2011

The challenge this proposal addresses is to speed up the development of concurrent programs by increasing the rate at which these programs evolve. The goal of this proposal is to generate methods and tools that help software engineers increase confidence in the correct operation of their programs. To achieve this goal, this proposal advocates testing of concurrent software using a systematic approach capable

continued on page 16

continued from page 15

of enumerating possible executions of a concurrent program. The systematic testing approach presented by this proposal is enabled by an apparatus which repeatedly executes a program test while exploring different orders in which concurrent events in the program could occur.

Novel features of the apparatus include: 1) its ability to control a program without the need make any changes to the program, 2) its language and architecture agnostic nature which makes it suitable for systematic testing of a wide range of concurrent systems, 3) its internal use of a formal language for modeling program coordination which enables the apparatus to build a run-time abstraction of a test execution, to check not only for errors that occur, but also for errors that might have occurred, and to provide a theoretical foundation for state space reduction, and 4) its ability to quickly and efficiently enumerate different test executions using a parallel algorithm, which enables concurrent exploration of hundreds of different test executions, coupled with a novel hierarchical partial order reduction scheme, which avoids exploration of equivalent behaviors and thus reduces the state space to be explored by orders of magnitude.

MASTERS THESIS:
Landslide: Systematic Dynamic Race Detection in Kernel-space

Ben Blum, SCS

May 2012

Carnegie Mellon University School of Computer Science Technical Report CMU-CS-12-118, May 2012.

Systematic exploration is an approach to finding race conditions by deterministically executing every possible interleaving of thread transitions and identifying which ones expose bugs. Current systematic exploration techniques are suitable for testing user-space programs, but are inadequate for testing kernels, where the testing framework's control over concurrency is more complicated.

Jiří Šimša presents his research on "Efficient Exploratory Testing of Concurrent Systems" at the 2011 PDL Retreat.

We present Landslide, a systematic exploration tool for finding races in kernels. Landslide targets Pebbles, the kernel specification that students implement in the undergraduate Operating Systems course at Carnegie Mellon University (15-410). We discuss the techniques Landslide uses to address the general challenges of kernel-level concurrency, and we evaluate its effectiveness and usability as a debugging aid. We show that our techniques make systematic testing in kernel-space feasible, and that Landslide is a useful tool for doing so in the context of 15-410.

MASTERS THESIS:
A Statistical Study for File System Metadata On High Performance Computing Sites

Yifan Wang , INI

April 2012

High performance parallel file systems are critical to the performance of super computers, are specialized to provide different computing services and are changing rapidly in both hardware and software, whose unusual access pattern has drawn great research interest. Yet little knowledge of how file systems evolve and how the way people use file systems change is known, even though significant effort and money has been put into upgrading storage device and designing new file systems. In this paper, we report on the statistics of supercomputing file systems from Parallel Data Lab (PDL) and Los Alamos National Lab (LANL) and compare the current data against their statistics 4 years ago to discover changes in technology and usage pattern and to observe new interesting characters.

MASTERS THESIS:
End-to-end Tracing in HDFS

William Wang , SCS

July 2011

Carnegie Mellon University School of Computer Science Technical Report CMU-CS-11-120, July 2011.

Debugging performance problems in distributed systems is difficult. Thus many debugging tools are being developed to aid diagnosis. Many of the most interesting new tools require information from end-to-end tracing in order to perform their analysis. This paper describes the development of an end-to-end tracing framework for the Hadoop Distributed File System. The approach to instrumentation in this implementation differs from previous ones as it focuses on detailed low-level instrumentation. Such instrumentation encounters the problems of large request flow graphs and a large number of different kinds of graphs, impeding the effectiveness of the diagnosis tools that use them. This report describes how to instrument at a fine granularity and explain techniques to handle the resulting challenges. The current implementation is evaluated in terms of performance, scalability, the data the instrumentation generates, and its ability to be used to solve performance problems.

continued from page 1

associate activity records with individual requests by propagating a per-request identifier, which is stored within the record. Activity records can be stitched together, either offline or online, to yield request-flow graphs, which show the control flow of individual requests.

Our technique of comparing request flows between two periods identifies distribution changes in request-flow behaviour and ranks them according to their contribution to identify the most important differences between two sets. Conversely, anomaly detection techniques mine a single period's request flows to identify rare ones that differ greatly from others.

We have implemented request-flow comparison in a toolset called Spectroscope and used it to diagnose performance problems observed in Ursa Minor, a distributed storage service, and in certain Google services. Spectroscope is not designed to replace developers; rather it is intended to serve as an important step in the workflow they use to diagnose problems. Sometimes, it can help developers identify the root cause immediately, or at least localize the problem to a specific area of the system. In other cases, it can help eliminate the distributed system as the root cause by verifying that its behaviour has not changed, allowing developers to focus their efforts on external factors. By describing several real problems, we illustrate the utility of comparing request flows and show that our algorithms enable effective use of this technique.

Spectroscope: Categorizing and Comparing Request Flows

Even small distributed systems can service hundreds to thousands of requests per second, so comparing all of them is not feasible. Instead, exploiting a general expectation that requests that take the same path should incur similar costs, Spectroscope groups identically-structured requests into unique categories and uses them as the basic unit for comparing request flows. For example, requests whose structures are identical because they hit in a NFS server's data and metadata cache will be grouped into the same category, whereas requests that miss in both will be grouped in a different one. Requests are deemed structurally identical if their string representations, as determined by a depth-first traversal, are identical. For requests with parallel substructures, Spectroscope computes all possible string representations when determining which category to place them in.

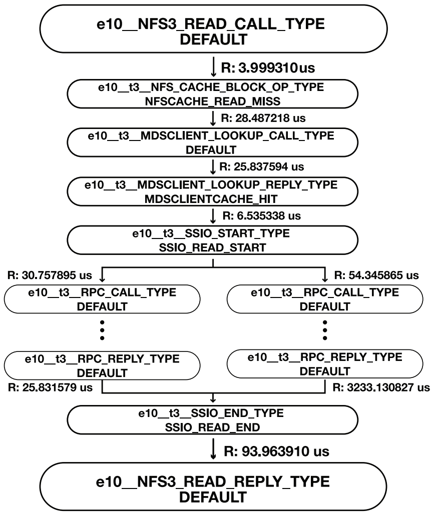


Figure 2: Example request-flow graph. The graph shows a striped READ in the Ursa Minor distributed storage system. Nodes represent trace points and edges are labeled with the time between successive events. Parallel substructures show concurrent threads of activity. Node labels are constructed by concatenating the machine name (e.g., e10), component name (e.g., NFS3), trace-point name (e.g., READ CALL TYPE), and an optional semantic label (e.g., NFSCACHE READ MISS). Due to space constraints, trace points executed on other components as a result of the NFS server's RPC calls are not shown.

For each category, Spectroscope identifies aggregate statistics, including request count, average response time, and variance. To identify where time is spent, it also computes average edge latencies and corresponding variances.

Performance changes often manifest as changes in how requests are serviced. The non-problem period (before the change) and the problem period (after the change), usually reveal some changes in the observed request flows. New request flows in the problem period are referred to as mutations and request flows corresponding to how they were serviced in the non-problem period as precursors. Response time mutations correspond to requests that have increased only in cost between the periods; their precursors are requests that exhibit the same structure, but whose response time is different. Structural mutations correspond to requests that take different paths through the system in the problem period. Identifying mutations helps localize sources of change and gives insight into their effects.

When comparing request flows, Spectroscope takes as input request-flow graphs from the non-problem period and the problem period. Categories are created composed of requests from both periods, and statistical tests and heuristics are used to identify which contain structural mutations, response-time mutations, or precursors. Categories containing mutations are presented to the developer in a list ranked by expected contribution to the performance change. Spectroscope is also able to provide further insight into performance changes by identifying the low-level parameters (e.g., client parameters or function call parameters) that best differentiate a chosen mutation and its precursors. For example, in Ursa Minor, one performance slow-down, which manifested many structural mutations, was caused by a change in a parameter sent by the client. For problems like this, highlighting the specific low-level differences can immediately identify the root cause.

Ursa Minor Case Studies

Like most modern scalable distributed

continued on page 18

continued from page 17

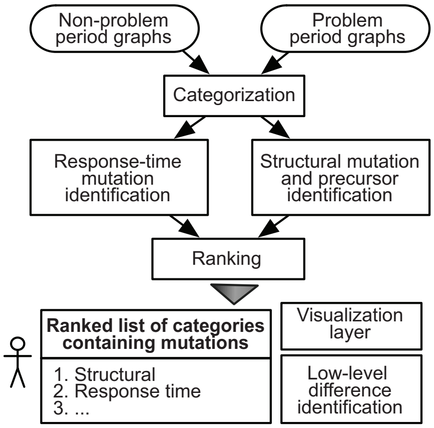

Non-problem period graphs
Problem period graphs
Categorization
Response-time mutation identification
Structural mutation and precursor identification
Ranking
Ranked list of categories containing mutations
1. Structural
2. Response time
3. ...
Visualization layer
Low-level difference identification

Figure 3. Spectroscope's workflow for comparing request flows. First, Spectroscope groups requests from both periods into categories. Second, it identifies which categories contain mutations or precursors. Third, it ranks mutation categories according to their expected contribution to the performance change. Developers are presented this ranked list. Visualizations of mutations and their precursors can be shown. Also, low-level differences can be identified for them.

storage systems, Ursa Minor separates metadata services from data services, such that clients can access data on storage nodes without moving it all through metadata servers. An Ursa Minor instance (called a "constellation") consists of potentially many NFS servers (for unmodified clients), storage nodes (SNs), metadata servers (MDSs), and end-to-end-trace servers. To access data, clients send a request to a metadata server asking for the appropriate permissions and locations of the data on the storage nodes and then the storage nodes are accessed directly. The components of Ursa Minor are usually run on separate machines within a datacenter. Though Ursa Minor supports an arbitrary number of components, our experiments used a simple five-machine configuration: one NFS server, one metadata server, one trace server, and two storage nodes. One storage node stores data, while the other stores metadata. Ursa Minor's tracing infrastructure, Stardust, uses request sampling to capture trace data for a subset of entire requests, with per-request decisions made randomly when a request enters the system. Ursa Minor contains approximately 200 trace points, 124 manually inserted as well as automatically generated ones for each RPC send and receive function. The start and end of concurrent threads of activity are identified using explicit split and join trace points.

Several case studies undertaken on the Ursa Minor platform used Spectroscope to diagnose and solve known performance problems by comparing request flows, proving its effectiveness at identifying the root causes of the problems. The experiments confirmed that comparing request flows helps developers diagnose performance changes caused by modifications to the system configuration. Many distributed systems contain large configuration files with esoteric parameters that, if modified, can result in perplexing performance changes. Spectroscope was able to show how various configuration options affect system behavior. Comparing request flows can also help developers identify performance problems that arise due to a workload change by highlighting relevant low-level parameter differences such as I/O size.

One case study demonstrated how comparing request flows can help developers account for unexpected performance loss when adding new features. Here, the problem was due to unanticipated contention several layers of abstraction below the feature addition.

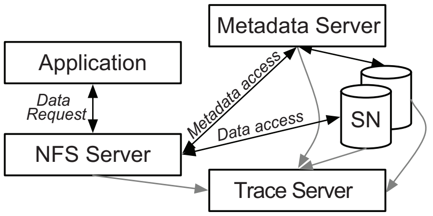


Figure 4. Ursa Minor Architecture. Ursa Minor can be deployed in many configurations, with an arbitrary number of NFS servers, metadata servers, storage nodes (SNs), and trace servers. Here, a simple five-component configuration is shown.

To improve performance by prefetching metadata to clients on every mandatory metadata server access, in hopes of minimizing the total number of accesses necessary, server-driven metadata prefetching was added to Ursa Minor. However, when implemented, this feature provided no improvement and a reason could not be found. Spectroscope instrumentation was added around the prefetching function and request-flow comparison used to expose surplus database accesses, which helped determine why server-driven metadata prefetching did not improve performance: the extra time spent in DB calls by metadata server accesses outweighed the time savings generated by the increase in client cache hits.

A second case study showed how request-flow comparison can be used to diagnose performance degradations over time, in this case due to a long-lived design problem. Though simple counters could have shown that CREATEs were very expensive, they would not have shown that the root cause was excessive metadata server/storage node interaction. Spectroscope was used to compare request flows between the first 1,000 CREATEs issued and the last 1,000 during a benchmark run. Spectroscope's results showed categories that contained both structural and response-time mutations. The response-time mutations were the expected result of data structures in the NFS server and metadata server whose performance decreased linearly with load. Analysis of the structural mutations, however, revealed two architectural issues that accounted for the degradation.

In a third case study, a synthetic problem was injected into Ursa Minor to show how request-flow comparison can be used to diagnose slowdowns due to feature additions or regressions and to assess Spectroscope's sensitivity

continued on page 19

continued from page 18

to changes in response time. Problem period runs included a spin loop injected into the storage nodes' WRITE codepath. Any WRITE request that accessed a storage node incurred this extra delay. Spectroscope was able to identify the resulting response-time mutations and localize them to the affected edges.

A fourth case study, recreating a real problem observed in our Ursa Minor deployment, showed that it is also possible to use the results of Spectroscope as proof that nothing within a distributed system has changed. This frees the developers to focus their efforts on external factors that may be causing problems.

For more information on Spectroscope's internals and more case studies (including some exploration in Google's environment), please see [1].

References

[1] Raja R. Sambasivan, Alice X. Zheng, Michael De Rosa, Elie Krevat, Spencer Whitman, Michael Stroucken, William Wang, Lianghong Xu, Gregory R. Ganger. Diagnosing Performance Changes by Comparing Request Flows. 8th USENIX Symp. on Networked Systems Design and Implementation (NSDI'11). March 30 - April 1, 2011. Boston, MA.

Mark your calendars for the 20th Annual Parallel Data Lab Workshop and Retreat, to be held at Bedford Springs Resort in Bedford Springs, PA, from November 5 to 7, 2012.

BIG DATA STORAGE

Garth Gibson

In the past few years PDL has been working to better understand the deep differences between scaling storage systems for the enterprise, for supercomputing and for internet services. Today it appears that all three are headed toward Big Data systems. Pragmatically there will be Big Data systems that solve problems very similar to perhaps only one of these original workloads, but betting odds appear to point to a convergence around analytics. Whether it be enterprises analyzing business records and customer databases to gain competitive or efficiency advantage, science processing physical sensor logs or simulation outputs to discover previously unrecognized patterns, governments processing public data to detect and intercede on yet unrecognized cyber and physical threats, or internet services collecting and analyzing social network databases to understand people and develop personalized advertising for them, vast amounts of captured data will be statistically and continuously analyzed. Traditional data storage systems are suspect of being insufficient to the demands of scale, and new storage systems have been developed to specialize to narrow but demanding problems.

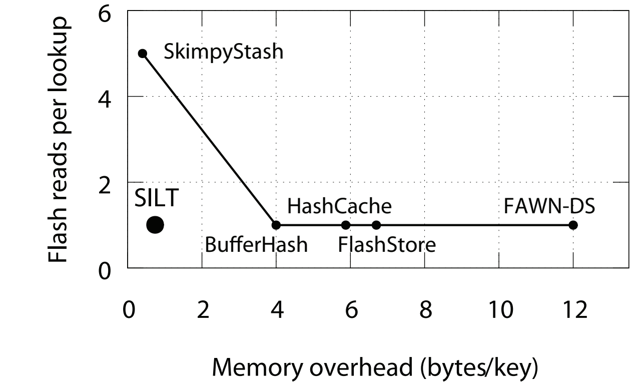


Figure 1. The memory overhead and lookup performance of SILT and the recent key-value stores. For both axes, smaller is better.

PDL researchers have added RAID to Hadoop HDFS, to reduce capacity overheads from 200% to 25% and along the way understand better the role of replication in making computation faster [1]. They have developed power efficient key-value stores, exploiting the right processing and SSD technology for the job at hand, winning power-efficient sort competitions and along the way understanding the role of denser indexing of stored data for fast access of small things [2, 3]. They have deconstructed data-intensive distributed file systems and compared fault models, performance and semantics to more traditional parallel file systems in use in supercomputing, and found the fault models and redundancy strategies to be the most significant difference [4]. They have de-aggregated highly concurrently modified huge data structures such as giant directories and checkpoints of parallel applications, using interposition layers to split out components into less concurrently modified data structures and load balanced data components over storage systems [5, 6].

In general we see that large, sequentially processed data scales pretty well, but large collections of small data, non-sequentially and concurrently accessed, can be quite a bit harder. For example, essentially none of the widely used parallel or data-intensive file

continued on page 20

continued from page 19

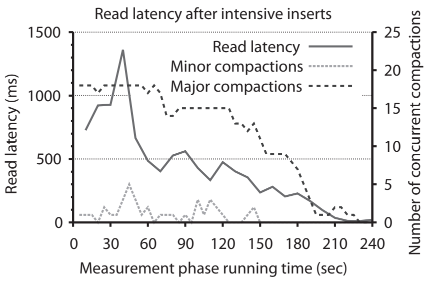


Figure 2. Understanding read latencies in Accumulo after ingest-intensive workloads and the correlations with compactions on its servers.

systems have metadata subsystems that arbitrarily scale for most workloads.

Looking to scaling storage services for small, concurrent data access, we extended a cloud database benchmarking suite, YCSB++, to explore advanced features—eventual consistency, server side filtering and bulk insert, for example [7] (Figure 1) One of our motivations for extending this benchmark was to assist the emergence of a new open source version of Big Data, largely built by NSA researchers. Called Accumulo (accumulo.apache.org), this system features fine grained access control and server-side programming of mutations, as well as pluggable policy managers, and powerful ingestion and data management optimizations. During our collaboration Accumulo joined the Apache incubator and graduated to be a top level project.

One aspect of key-value stores like Accumulo and Big Table is the use of Log-Structured Merge (LSM) trees—on-disk immutable B-trees of recent changes in the data in one leaf of the overall B-tree that are searched when a reference misses in the in-memory cache. As data is added and changes made, the number of these logs increases, wasting space with inserted records that have also been deleted, but more importantly forcing more disk accesses on look ups that miss in memory and might be in one of multiple LSM tree files. Asynchronous compaction merges LSM tree files to filter out dead records and reduces the number of places a data item might be. One of the more interesting aspects of key-value stores based on LSM trees is the amount of work that is done in these compactions and the implications on user performance. Figure 2 shows a four minute window of time after a long intensive period of adding data to Accumulo during which is light random read and update workload "measures" latency during compaction—large read response times persist for most of the three minutes it takes the compaction process to "clean up" after the intensive insertion is over. We expect future work will concentrate on asynchronous work like compaction, lowering its impact on user work and improving its efficiency.

Going forward we are very interested in improving the scalability of metadata in all Big Data storage systems, especially the internal metadata, because this has long been poorly done in almost all deployed storage systems. Some of the techniques of LSM tree representation are already showing good results for improving the efficiency of metadata storage inside a file system—see Kai Ren's poster at Visit Day.

References

[1] Bin Fan, Wittawat Tantisiriroj, Lin Xiao, Garth Gibson. DiskReduce: Replication as a Prelude to Erasure Coding in Data-Intensive Scalable Computing. Carnegie Mellon Univsersity Parallel Data Laboratory Technical Report CMU-PDL-11-112, October, 2011.

[2] Hyeontaek Lim, Bin Fan, David Andersen and Michael Kaminsky. SILT: A Memory-Efficient, High-Performance Key-Value Store. ACM Symposium on Operating Systems Principles (SOSP'11), Cascais, Portugal, October 2011.

[3] David Andersen, Jason Franklin, Michael Kaminsky, Amar Phanishayee, Lawrence Tan, Vijay Vasudevan. FAWN: A Fast Array of Wimpy Nodes. Proc. 22nd ACM Symposium on Operating Systems Principles (SOSP 2009), Big Sky, MT. October 2009.

[4] Wittawat Tantisiriroj, Swapnil Patil, Garth Gibson, Seung Woo Son, Samuel J. Lang, Robert B. Ross. On the Duality of Data-intensive File System Design: Reconciling HDFS and PVFS. SC11, November 12-18, 2011, Seattle, Washington USA.

[5] Swapnil Patil, Garth Gibson. Scale and Concurrency of GIGA+: File System Directories with Millions of Files. Proc. of the 9th USENIX Conference on File and Storage Technologies (FAST '11), San Jose CA, February 2011.

[6] John Bent, Garth Gibson, Gary Grider, Ben McClelland, Paul Nowoczynski, James Nunez, Milo Polte, Meghan Wingate. PLFS: A Checkpoint Filesystem for Parallel Applications. SC09, November 15, 2009. Portland, Oregon.

[7] Swapnil Patil, Milo Polte, Kai Ren, Wittawat Tantisiriroj, Lin Xiao, Julio López, Garth Gibson. YCSB++: Benchmarking and Performance Debugging of Advanced Features in Scalable Table Stores. Proc. of the 2nd ACM Symposium on Cloud Computing (SOCC '11), October 27–28, 2011, Cascais, Portugal. August 2011.

Vanish Talwar (HP Labs), Greg Ganger, and Michael Kozuch (Intel) discussing cloud computing at the 2011 PDL Retreat.

continued from page 4

Big Effect: Provable Load Balancing for Randomly Partitioned Cluster Services" at SOCC'11 in Cascais, Portugal.

September 2011

- Garth gave a SNIA Storage Developer Conference (SDC11) keynote: "Scalable Table Stores: Tools for Understanding Advanced Key-Value Systems for Hadoop."
- Fan Guo presented his Ph.D. dissertation "Mining and Querying Multimedia Data."
- Matthew Wachs defended his Ph.D. dissertation "Performance Insulation: More Predictable Shared Storage."
- Lei Li defended his Ph.D. dissertation on "Fast Algorithms for Mining Co-evolving Time Series."
- Garth presented a state of "Storage Systems" at the Oak Ridge National Lab Discovery 2015 Conference, Oak Ridge, TN.

August 2011

- The Intel Science and Technology Centers (ISTC) for Cloud Computing Research and for Embedded Computing both opened at CMU.
- Greg Ganger and Garth Gibson participated in the High End Computing File Systems and I/O Workshop in Arlington, VA.
- While there, Garth also took part in "Death of Disks Panel: A Darwinian Evolution; Principles of Operation for Shingled Disk Devices", and presented "YCSB++: Benchmarking Cloud DBs;" Greg participated in a panel on Storage QoS.

July 2011

- Satya (M. Satyanarayanan) received the SIGMOBILE 2010 Outstanding Contributions Award at Mobisys'11.
- Anshul Gandhi presented "Minimizing Data Center SLA Violations and Power Consumption via Hybrid Resource Provisioning" at the second IEEE International Green Computing Conference in Orlando. FL.
- Jiří Šimša presented "dBug: Systematic Testing of Distributed and Multi-threaded Systems" at the 18th International Workshop on Model Checking of Software (SPIN'11) in Snowbird, UT.
- Garth presented "BigData Storage Systems: Large Datasets in Astrophysics and Cosmology," at the Institute for Computing in Science (ICiS 2011), Park City, UT.
- Julio López presented "Recipes for Baking Black Forest Databases: Building and Querying Black Hole Merger Trees from Cosmological Simulations," 23rd Scientific and Statistical Database Management Conference (SSDBM'11).

June 2011

- Anshul Gandhi presented "Distributed, Robust Auto-Scaling Policies for Power Management in Compute Intensive Server Farms" at the Open Cirrus Summit in Moscow, Russia.
- Onur Mutlu received the IEEE Young Computer Architect Award in recognition of his contributions to research and education in the computer architecture field.
- The FAWN Team was the winner of the 2011 10GB JouleSort Daytona and Indy Categories.
- A team including several PDL Alums (Ippokratis Pandis, Ryan Johnson and Anastasia Ailamaki) won the award for best demo at SIGMOD 2011 for their work on "A Data-oriented Transaction Execution Engine and Supporting Tools."
- Swapnil Patil received an ACM student award for his development of a file system director service that scales to millions of files.
- Amar Phanishayee proposed his thesis research topic "One Key-value System to Store Them All!"
- William Wang presented his Masters Thesis "End-to-end Tracing in HDFS."
- Kai Ren presented "Otus: Resource Attribution and Metrics Correlation in Data-Intensive Clusters," to the The 2nd International Workshop on MapReduce and its Applications (MapReduce'11), San Jose, CA.

May 2011

- Ippokratis Pandis defended his Ph.D. thesis "Scalable Transaction Processing through Data-oriented Execution."
- Vijay Vasudevan presented "The Case for VOS: The Vector Operating System" at the 13th Workshop on Hot Topics in Operating Systems (HotOS 2011) in Napa, CA.

April 2011

- Jiří Šimša proposed "Systematic and Scalable Testing of Concurrent Systems" as the subject of his thesis research.
- 13th Annual PDL Spring Industry Visit Day.

Mike Kasick presents "Black-Box Localization of Storage Problems in Parallel File Systems."

continued from page 7

Efficient Exploratory Testing of Concurrent Systems

Jiří Šimša, Randy Bryant, Garth Gibson & Jason Hickey

Carnegie Mellon University Parallel Data Laboratory Techical Report CMU-PDL-11-113, November 2011.

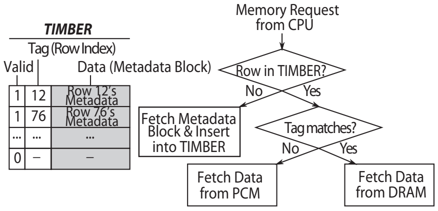
In our experience, exploratory testing has reached a level of maturity that makes it a practical and often the most cost-effective approach to testing. Notably, previous work has demonstrated that exploratory testing is capable of finding bugs even in well-tested systems [4, 17, 24, 23]. However, the number of bugs found gives little indication of the efficiency of a testing approach. To drive testing efficiency, this paper focuses on techniques for measuring and maximizing the coverage achieved by exploratory testing. In particular, this paper describes the design, implementation, and evaluation of Eta, a framework for exploratory testing of multithreaded components of a large-scale cluster management system at Google. For simple tests (with millions to billions of possible executions), Eta achieves complete coverage one to two orders of magnitude faster than random testing. For complex tests, Eta adopts a state space reduction technique to avoid the need to explore over 85% of executions and harnesses parallel processing to explore multiple test executions concurrently, achieving a throughput increase of up to 17.5X.

Enabling Efficient and Scalable Hybrid Memories Using Fine-Granularity DRAM Cache Management

Justin Meza, Jichuan Chang, HanBin Yoon, Onur Mutlu & Parthasarathy Ranganathan

IEEE Computer Architecture Letters (CAL), May 2012.

Hybrid main memories composed of DRAM as a cache to scalable non-volatile memories such as phase-change memory (PCM) can provide much larger storage capacity than traditional main memories. A key challenge for enabling high-performance and scalable hybrid memories, though, is efficiently managing the metadata (e.g., tags) for data cached in DRAM at a fine granularity. Based on the observation that storing metadata off-chip in the same row as their data exploits DRAM row buffer locality, this paper reduces the overhead of fine-granularity DRAM caches by only caching the metadata for recently accessed rows on-chip using a small buffer. Leveraging the flexibility and efficiency of such a fine-granularity DRAM cache, we also develop an adaptive policy to choose the best granularity when migrating data into DRAM. On a hybrid memory with a 512MB DRAM cache, our proposal using an 8KB on-chip buffer can achieve within 6% of the performance of, and 18% better energy efficiency than, a conventional 8MB SRAM metadata store, even when the energy overhead due to large SRAM metadata storage is not considered.



TIMBER (a buffer for recently-accessed metadata blocks) organization and metadata lookup.

Minimizing Data Center SLA Violations and Power Consumption via Hybrid Resource Provisioning

Anshul Gandhi,Yuan Chen, Daniel Gmach, Martin Arlitt & Manish Marwah

2nd IGCC 2011 (IEEE International Green Computing Conference 2011) July 25-28, 2011 Orlando, Florida, USA.

This paper presents a novel approach to correctly allocate resources in data centers, such that SLA violations and energy consumption are minimized. Our approach first analyzes historical workload traces to identify long-term patterns that establish a "base" workload. It then employs two techniques to dynamically allocate capacity: predictive provisioning handles the estimated base workload at coarse time scales (e.g., hours or days) and reactive provisioning handles any excess workload at finer time scales (e.g., minutes). The combination of predictive and reactive provisioning achieves a significant improvement in meeting SLAs, conserving energy, and reducing provisioning costs. We implement and evaluate our approach using traces from four production systems. The results show that our approach can provide up to 35% savings in power consumption and reduce SLA violations by as much as 21% compared to existing techniques, while avoiding frequent power cycling of servers.

Distributed, Robust Auto-Scaling Policies for Power Management in Compute Intensive Server Farms

Anshul Gandhi, Mor Harchol-Balter, Ram Raghunathan & Michael A. Kozuch

5th International Open Cirrus Summit. June 01 – 03, 2011, Moscow, Russia.

Server farms today often over-provision resources to handle peak demand, resulting in an excessive waste of power. Ideally, server farm capacity should be dynamically adjusted based on the incoming demand. However, the unpredictable and time-varying nature of customer demands makes it very difficult to efficiently scale capacity in server farms. The problem is further exacerbated by the large setup time needed to increase capacity, which can adversely impact response times as well as utilize additional power.

In this paper, we present the design and implementation of a class of

continued on page 23

continued from page 22

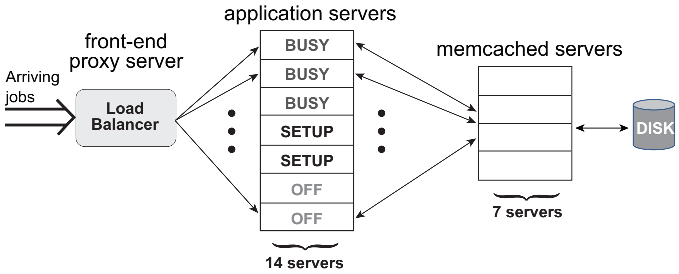
Distributed and Robust Auto-Scaling policies (DRAS policies), for power management in compute intensive server farms. Results indicate that the DRAS policies dynamically adjust server farm capacity without requiring any prediction of the future load, or any feedback control. Implementation results on a 21 server test-bed show that the DRAS policies provide near-optimal response time while lowering power consumption by about 30% when compared to static provisioning policies that employ a fixed number of servers.

The Case for Sleep States in Servers

Anshul Gandhi & Mor Harchol-Balter

HotPower'11, October 23, 2011, Cascais, Portugal.

While sleep states have existed for mobile devices and workstations for some time, these sleep states have largely not been incorporated into the servers in today's data centers. Chip designers have been unmotivated to design sleep states because data center administrators haven't expressed any desire to have them. High setup times make administrators fearful of any form of dynamic power management, whereby servers are suspended or shut down when load drops. This general reluctance has stalled research into whether there might be some feasible sleep state (with sufficiently low setup overhead and/or sufficiently low power) that would actually be beneficial in data centers. This paper uses both experimentation and theory to investigate the regime of sleep states that should be advantageous in data centers. Implementation experiments involve a 24-server multi-tier testbed, serving a web site of the type seen in Facebook or Amazon with key-value workload and a range of hypothetical sleep states. Analytical modeling is used to understand the effect of scaling up to larger data centers. The goal of this research is to encourage data center administrators to consider dynamic power management and to spur chip designers to develop useful sleep states for servers.



Sleep state in servers: experimental setup.

Bottleneck Identification and Scheduling in Multithreaded Applications

José A. Joao, M. Aater Suleman, Onur Mutlu & Yale N. Patt

17th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), London, UK, March 2012.

Performance of multithreaded applications is limited by a variety of bottlenecks, e.g. critical sections, barriers and slow pipeline stages. These bottlenecks serialize execution, waste valuable execution cycles, and limit scalability of applications. This paper proposes Bottleneck Identification and Scheduling (BIS), a cooperative software-hardware mechanism to identify and accelerate the most critical bottlenecks. BIS identifies which bottlenecks are likely to reduce performance by measuring the number of cycles threads have to wait for each bottleneck, and accelerates those bottlenecks using one or more fast cores on an Asymmetric Chip Multi- Processor (ACMP). Unlike previous work that targets specific bottlenecks, BIS can identify and accelerate bottlenecks regardless of their type. We compare BIS to four previous approaches and show that it outperforms the best of them by 15% on average. BIS' performance improvement increases as the number of cores and the number of fast cores in the system increase.

Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms

Patrick Gage Kelley, Saranga Komanduri, Michelle L. Mazurek, Rich Shay, Tim Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor & Julio López

2012 IEEE Symposium on Security and Privacy, May 2012.

Text-based passwords remain the dominant authentication method in computer systems, despite significant advancement in attackers' capabilities to perform password cracking. In response to this threat, password composition policies have grown increasingly complex. However, there is insufficient research defining metrics to characterize password strength and using them to evaluate password-composition policies. In this paper, we analyze 12,000 passwords collected under seven composition policies via an online study. We develop an efficient distributed method for calculating how effectively several heuristic password-guessing algorithms guess passwords. Leveraging this method, we investigate (a) the resistance of passwords created under different conditions to guessing; (b) the performance of guessing algorithms under different training sets; (c) the relationship between passwords explicitly created under a given composition policy and other passwords that happen to meet the same requirements; and (d) the relationship between guessability, as measured with password-cracking

continued on page 24

continued from page 23

algorithms, and entropy estimates. Our findings advance understanding of both password-composition policies and metrics for quantifying password security.

Tag, You Can See It! Using Tags for Access Control in Photo Sharing

Peter F. Klemperer, Yuan Liang, Michelle L. Mazurek, Manya Sleeper, Blase Ur, Lujo Bauer, Lorrie Faith Cranor, Nitin Gupta & Michael K. Reiter

CHI 2012: Conference on Human Factors in Computing Systems, May 2011.

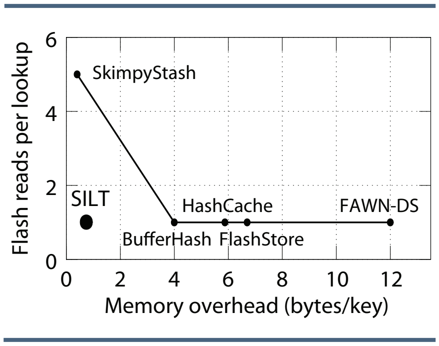
Users often have rich and complex photo-sharing preferences, but properly configuring access control can be difficult and time-consuming. In an 18-participant laboratory study, we explore whether the keywords and captions with which users tag their photos can be used to help users more intuitively create and maintain access-control policies. We find that (a) tags created for organizational purposes can be repurposed to create efficient and reasonably accurate access-control rules; (b) users tagging with access control in mind develop coherent strategies that lead to significantly more accurate rules than those associated with organizational tags alone; and (c) participants can understand and actively engage with the concept of tag-based access control.

SILT: A Memory-Efficient, High-Performance Key-Value Store

Hyeontaek Lim, Bin Fan, David Andersen & Michael Kaminsky

ACM Symposium on Operating Systems Principles (SOSP'11), Cascais, Portugal, October 2011.

SILT (Small Index Large Table) is a memory-efficient, high-performance key-value store system based on flash storage that scales to serve billions of key-value items on a single node. It requires only 0.7 bytes of DRAM per entry and retrieves key/value pairs using on average 1.01 flash reads each. SILT combines new algorithmic and systems techniques to balance the use of memory, storage, and computation. Our contributions include: (1) the design of three basic key-value stores each with a different emphasis on memory-efficiency and write-friendliness; (2) synthesis of the basic key-value stores to build a SILT key-value store system; and (3) an analytical model for tuning system parameters carefully to meet the needs of different workloads. SILT requires one to two orders of magnitude less memory to provide comparable throughput to current high-performance key-value systems on a commodity desktop system with flash storage.



The memory overhead and lookup performance of SILT and the recent key-value stores. For both axes, smaller is better.

YCSB++: Benchmarking and Performance Debugging Advanced Features in Scalable Table Stores

Swapnil Patil, Milo Polte, Kai Ren, Wittawat Tantisiriroj, Lin Xiao, Julio Lopez, Garth Gibson, Adam Fuchs & Billie Rinaldi

Proc. of the 2nd ACM Symposium on Cloud Computing (SOCC '11), October 27–28, 2011, Cascais, Portugal.

Inspired by Google's BigTable, a variety of scalable, semistructured, weak-semantic table stores have been developed and optimized for different priorities such as query speed, ingest speed, availability, and interactivity. As these systems mature, performance benchmarking will advance from measuring the rate of simple workloads to understanding and debugging the performance of advanced features such as ingest speed-up techniques and function shipping filters from client to servers. This paper describes YCSB++, a set of extensions to the Yahoo! Cloud Serving Benchmark (YCSB) to improve performance understanding and debugging of these advanced features. YCSB++ includes multi- tester coordination for increased load and eventual consistency measurement, multi-phase workloads to quantify the consequences of work deferment and the benefits of anticipatory configuration optimization such as B-tree pre-splitting or bulk loading, and abstract APIs for explicit incorporation of advanced features in benchmark tests. To enhance performance debugging, we customized an existing cluster monitoring tool to gather the internal statistics of YCSB++, table stores, system services like HDFS, and operating systems, and to offer easy post-test correlation and reporting of performance behaviors. YCSB++ features are illustrated in case studies of two BigTable-like table stores, Apache HBase and Accumulo, developed to emphasize high ingest rates and fine-grained security.

Practical Experiences with Chronics Discovery in Large Telecommunications Systems

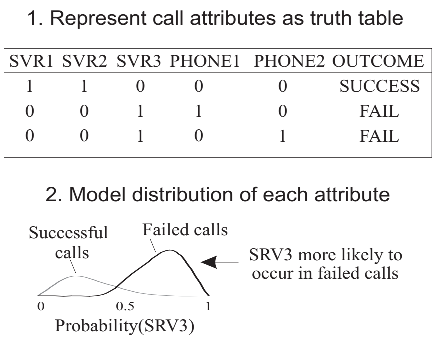
Soila P. Kavulya, Kaustubh Joshi, Matti Hiltunen , Scott Daniels (AT&T), Rajeev Gandhi & Priya Narasimhan

Workshop on System Logs and the Application of Machine Learning Techniques (SLAML), Cascais, Portugal, October 2011.

Chronics are recurrent problems that fly under the radar of operations teams because they do not perturb the system enough to set off alarms or violate

continued on page 25

continued from page 24



1. Represent call attributes as truth table

SVR1	SVR2	SVR3	PHONE1	PHONE2	OUTCOME
1	1	0	0	0	SUCCESS
0	0	1	1	0	FAIL
0	0	1	0	1	FAIL

2. Model distribution of each attribute

An overview of steps used by our top-down, statistical diagnosis algorithm.

service-level objectives. The discovery and diagnosis of never-before seen chronics poses new challenges as they are not detected by traditional threshold-based techniques, and many chronics can be present in a system at once, all starting and ending at different times. In this paper, we describe our experiences diagnosing chronics using server logs on a large telecommunications service. Our technique uses a scalable Bayesian distribution learner coupled with an information theoretic measure of distance (KL divergence), to identify the attributes that best distinguish failed calls from successful calls. Our preliminary results demonstrate the usefulness of our technique by providing examples of actual instances where we helped operators discover and diagnose chronics.

Don't Settle for Eventual: Scalable Causal Consistency for Wide-Area Storage with COPS

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

Proc. 23rd ACM Symposium on Operating Systems Principles (SOSP), Oct 2011.

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.

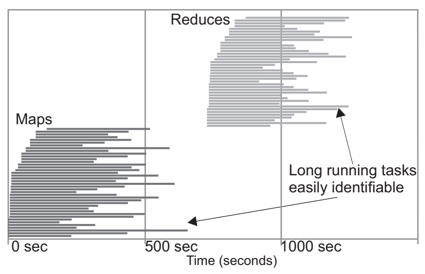
We present the design and implementation of COPS, a key-value store that delivers this consistency model across the wide-area. A key contribution of COPS is its scalability, which can enforce causal dependencies between keys stored across an entire cluster, rather than a single server like previous systems. 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.

Understanding and Improving the Diagnostic Workflow of MapReduce Users

Jason D. Campbell (Intel Labs Pittsburgh), Arun B. Ganesan, Ben Gotow, Soila P. Kavulya, James Mulholland, Priya Narasimhan, Sriram Ramasubramanian, Mark Shuster & Jiaqi Tan

ACM Symposium on Computer Human Interaction for Management of Information Technology (CHIMIT), Boston, MA, December 2011.

New abstractions are simplifying the programming of large clusters, but diagnosis nonetheless gets more and more challenging as cluster sizes grow: Debugging information increases linearly with cluster size, and the count of inter-component relationships grows quadratically. Worse, the new abstractions which simplified programming can also obscure the relationships between high-level (application) and low-level (task/process/disk/CPU) information flows. In this paper we analyze the workflow of several users and systems administrators connected with a large academic cluster (based the popular Hadoop implementation of the MapReduce abstraction) and propose improvements to the diagnosis-relevant information displays. We also offer a preliminary analysis of the efficacy of the changes we propose that demonstrates a 40% reduction in the time taken to accomplish 5 representative diagnostic tasks as compared to the current system.



Swimlane graph charting the start and end times, and durations of Map and Reduce tasks for a single job. The graph also highlights the inherent structure of MapReduce jobs with map tasks completing before reduce tasks.

Small Cache, Big Effect: Provable Load Balancing for Randomly Partitioned Cluster Services

Bin Fan, Hyeontaek Lim, David Andersen & Michael Kaminsky

ACM Symposium on Cloud Computing (SOCC'11), Cascais, Portugal, October, 2011.

Load balancing requests across a cluster of back-end servers is critical for avoiding performance bottlenecks

continued on page 26

continued from page 25

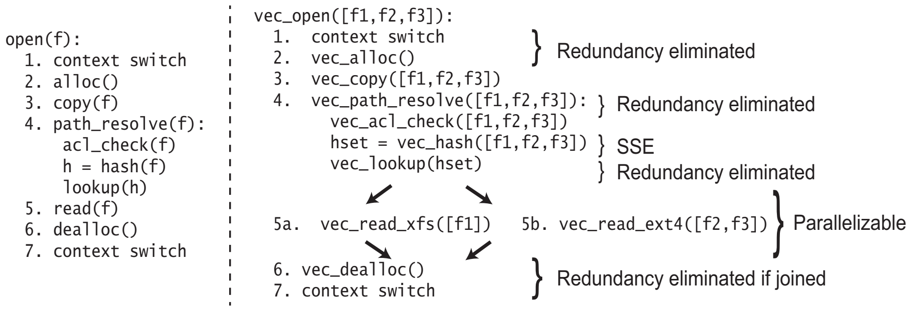
and meeting service-level objectives (SLOs) in large-scale cloud computing services. This paper shows how a small, fast popularity-based front-end cache can ensure load balancing for an important class of such services; furthermore, we prove an O(n log n) lower-bound on the necessary cache size and show that this size depends only on the total number of back-end nodes n, not the number of items stored in the system. We validate our analysis through simulation and empirical results running a key-value storage system on an 85-node cluster.

The Case for VOS: The Vector Operating System

Vijay Vasudevan, David Andersen & Michael Kaminsky

In 13th Workshop on Hot Topics in Operating Systems (HotOS 2011). May 2011.

Operating systems research for many-core systems has recently focused its efforts on supporting the scalability of OS-intensive applications running on increasingly parallel hardware. Lost amidst the march towards this parallel future is efficiency: Perfectly parallel software may saturate the parallel capabilities of the host system, but in doing so can waste hardware resources. This paper describes our motivation for the Vector OS, a design inspired by vector processing systems that provides efficient parallelism. The Vector OS organizes and executes requests for operating system resources through "vector" interfaces that operate on vectors of objects. We argue that these interfaces allow the OS to capitalize on numerous chances to both eliminate redundant work found in OS-intensive systems and use the underlying parallel hardware to its full capability, opportunities that are missed by existing operating systems.



```
open(f):
  1. context switch
  2. alloc()
  3. copy(f)
  4. path_resolve(f):
       acl_check(f)
       h = hash(f)
       lookup(h)
  5. read(f)
  6. dealloc()
  7. context switch
```

```
vec_open([f1,f2,f3]):
  1.  context switch
  2.  vec_alloc()
  3.  vec_copy([f1,f2,f3])
  4.  vec_path_resolve([f1,f2,f3]):
        vec_acl_check([f1,f2,f3])
        hset = vec_hash([f1,f2,f3])
        vec_lookup(hset)

  5a.  vec_read_xfs([f1])      5b. vec_read_ext4([f2,f3])

  6. vec_dealloc()
  7. context switch
```

Pseudocode for open() and proposed vec open(). vec open() provides opportunities for eliminating redundant code execution, vector execution when possible, and parallel execution otherwise.

Failure Diagnosis of Complex Systems

Soila P. Kavulya, Kaustubh Joshi, Felicita Di Giandomenico & Priya Narasimhan

In "*Resilience Assessment and Evaluation.*" Springer Verlag, 2011.

Failure diagnosis is the process of identifying the causes of impairment in a system's function based on observable symptoms, i.e., determining which fault led to an observed failure. Since multiple faults can often lead to very similar symptoms, failure diagnosis is often the first line of defense when things go wrong—a prerequisite before any corrective actions can be undertaken. The results of diagnosis also provide data about a system's operational fault profile for use in offline resilience evaluation. While diagnosis has historically been a largely manual process requiring significant human input, techniques to automate as much of the process as possible have significantly grown in importance in many industries including telecommunications, internet services, automotive systems, and aerospace. This chapter presents a survey of automated failure diagnosis techniques including both model-based and model-free approaches. Industrial applications of these techniques in the above domains are presented, and finally, future trends and open challenges in the field are discussed.

dBug: Systematic Testing of Distributed and Multi-threaded Systems

Jiří Šimša, Randy Bryant & Garth Gibson

18th International Workshop on Model Checking of Software (SPIN'11), Snowbird UT, July 2011.

In order to improve quality of an implementation of a distributed and multi-threaded system, software engineers inspect code and run tests. However, the concurrent nature of such systems makes these tasks challenging. For testing, this problem is addressed by stress testing, which repeatedly executes a test hoping that eventually all possible outcomes of the test will be encountered. In this paper we present the dBug tool, which implements an alternative method to stress testing called systematic testing. The systematic testing method implemented by dBug controls the order in which certain concurrent function calls occur. By doing so, the method can systematically enumerate possible inter- leavings of function calls in an execution of a concurrent system. The dBug tool can be thought of as a light-weight model checker, which uses the implementation of a distributed and multi-threaded system and its test as an implicit description of the state space to be explored. In this state space, the dBug tool performs a reachability analysis checking for a number of safety properties including the absence of 1) deadlocks, 2) conflicting non-reentrant function calls, and 3) system aborts and runtime assertions inserted by the user.

continued on page 27

continued from page 26

Time Series Clustering: Complex is Simpler!

Lei Li, B. Aditya Prakash

Proceedings of the 28th International Conference on Machine learning, June 28 - July 2, 2011, Bellevue, WA.

Given a motion capture sequence, how to identify the category of the motion? Classifying human motions is a critical task in motion editing and synthesizing, for which manual labeling is clearly inefficient for large databases. Here we study the general problem of time series clustering. We propose a novel method of clustering time series that can (a) learn joint temporal dynamics in the data; (b) handle time lags; and (c) produce interpretable features. We achieve this by developing complex-valued linear dynamical systems (CLDS), which include real-valued Kalman filters as a special case; our advantage is that the transition matrix is simpler (just diagonal), and the transmission one easier to interpret. We then present Complex- Fit, a novel EM algorithm to learn the parameters for the general model and its special case for clustering. Our approach produces significant improvement in clustering quality, 1.5 to 5 times better than well-known competitors on real motion capture sequences.

Exact and Approximate Computation of a Histogram of Pairwise Distances between Astronomical Objects.

Bin Fu, Eugene Fink, Garth Gibson & Jaime Carbonell

First Workshop on High Performance Computing in Astronomy (AstroHPC 2012), held in conjunction with the 21st International ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC 2012), June 18 or 19, 2012, Delft, the Netherlands.

We compare several alternative approaches to computing *correlation functions*, which is a cosmological application for analyzing the distribution of matter in the universe. This computation involves counting the pairs of galaxies within a given distance from each other and building a histogram that shows the dependency of the number of pairs on the distance.

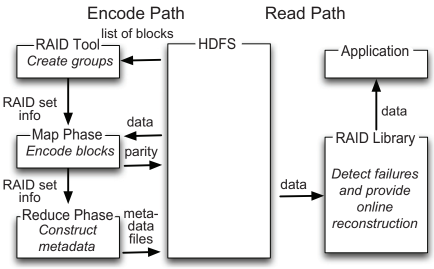
The straightforward algorithm for counting the exact number of pairs has the $O(n^2)$ time complexity, which is unacceptably slow for most astronomical and cosmological datasets, which include billions of objects. We analyze the performance of several alternative algorithms, including the exact computation with an $O(n^{5/3})$ average running time, an approximate computation with linear running time, and another approximate algorithm with sub-linear running time, based on sampling the given dataset and computing the correlation functions for the samples. We compare the accuracy of the described algorithms and analyze the tradeoff between their accuracy and running time. We also propose a novel hybrid approximation algorithm, which outperforms each other technique.

DiskReduce: Replication as a Prelude to Erasure Coding in Data-Intensive Scalable Computing

Bin Fan, Wittawat Tantisiriroj, Lin Xiao & Garth Gibson

Carnegie Mellon University Parallel Data Laboratory Technical Report CMU-PDL-11-112, October, 2011.

The first generation of Data-Intensive Scalable Computing file systems such as Google File System and Hadoop Distributed File System employed n replications for high data reliability, therefore delivering users only about 1/n of the total storage capacity of the raw disks. This paper presents DiskReduce, a framework integrating RAID into these replicated storage systems to significantly reduce the storage capacity overhead, for example, from 200% to 25% when triplicated data is dynamically replaced with RAID sets (e.g. 8 + 2 RAID 6 encoding). Based on traces collected from Yahoo!, Facebook and Opencloud cluster, we analyze (1) the capacity effectiveness of simple and not so simple strategies for grouping data blocks into RAID sets; (2) implication of reducing the number of data copies on read performance and how to overcome the degradation; and (3) different heuristics to mitigate "small write penalties." Finally, we introduce an implementation of our framework that has been built and submitted into the Apache Hadoop project.



Encode and read path for RAID files.

Switching the Optical Divide: Fundamental Challenges for Hybrid Electrical/Optical Datacenter Networks

Hamid Hajabdolali Bazzaz, Malveeka Tewari, Guohui Wang, George Porter, T. S. Eugene Ng, David G. Andersen, Michael Kaminsky, Michael A. Kozuch & Amin Vahdat

Proceedings of the 2nd ACM Symposium on Cloud Computing (SOCC), Oct 2011.

Recent proposals to build hybrid electrical (packet-switched) and optical (circuit switched) data center interconnects promise to reduce the cost, complexity, and energy requirements of very large data center networks. Supporting realistic traffic patterns, however, exposes a number of unexpected and difficult challenges to actually deploying these systems "in

continued on page 28

continued from page 27

the wild." In this paper, we explore several of these challenges, uncovered during a year of experience using hybrid interconnects. We discuss both the problems that must be addressed to make these interconnects truly useful, and the implications of these challenges on what solutions are likely to be ultimately feasible.

File System Virtual Appliances: Portable File System Implementations

Michael Abd-El-Malek, Matthew Wachs, James Cipar, Karan Sanghi, Gregory R. Ganger, Garth A. Gibson & Michael K. Reiter

ACM Transactions on Storage, Volume 8 Issue 3, April 12, 2012.

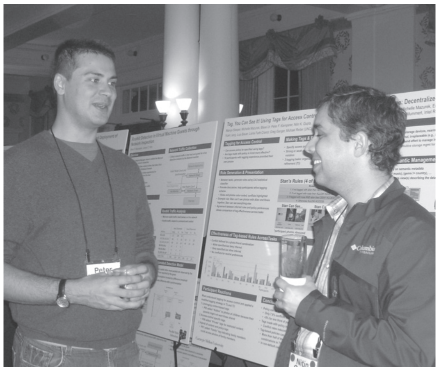
File system virtual appliances (FSVAs) address the portability headaches that plague file system (FS) developers. By packaging their FS implementation in a VM, separate from the VM that runs user applications, they can avoid the need to port the file system to each OS and OS version. A small FS-agnostic proxy, maintained by the core OS developers, connects the FSVA to whatever OS the user chooses. This paper describes an FSVA design that maintains FS semantics for unmodified FS implementations and provides desired OS and virtualization features, such as a unified buffer cache and VM migration. Evaluation of prototype FSVA implementations in Linux and NetBSD, using Xen as the VMM, demonstrates that the FSVA architecture is efficient, FS-agnostic, and able to insulate file system implementations from OS differences that would otherwise require explicit porting.

On the Duality of Data-intensive File System Design: Reconciling HDFS and PVFS

Wittawat Tantisiriroj, Swapnil Patil, Garth Gibson, Seung Woo Son, Samuel J. Lang & Robert B. Ross

Supercomputing 2011, November 12-18, 2011, Seattle, Washington USA.

Data-intensive applications fall into two computing styles: Internet services (cloud computing) or high-performance computing (HPC). In both categories, the underlying file system is a key component for scalable application performance. In this paper, we explore the similarities and differences between PVFS, a parallel file system used in HPC at large scale, and HDFS, the primary storage system used in cloud computing with Hadoop. We integrate PVFS into Hadoop and compare its performance to HDFS using a set of data-intensive computing benchmarks. We study how HDFS-specific optimizations can be matched using PVFS and how consistency, durability, and persistence tradeoffs made by these file systems affect application performance. We show how to embed multiple replicas into a PVFS file, including a mapping with a complete copy local to the writing client, to emulate HDFS's file layout policies. We also highlight implementation issues with HDFS's dependence on disk bandwidth and benefits from pipelined replication.



Peter Klemperer and Nitin Gupta discuss their research at a 2011 PDL Retreat poster session.

PDL Workshop and Retreat 2011.