PARALLEL DATA LABORATORY
CARNEGIE MELLON UNIVERSITY

PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • SPRING 2013
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
Samsung Information Systems America
Seagate Technology
STEC, Inc.
Symantec Corporation
VMware, Inc.
Western Digital

LazyTables: Faster Distributed Machine Learning through Staleness

Big Learning Group (participants listed at the end)

With the advent of "Big Data" sets and cheap multicore computers, distributed Machine Learning (ML) on clusters of multicore machines has long since passed from luxury to necessity. Yet, effective learning on big data goes beyond designing parallel algorithms that produce correct answers (or at least "good enough", as is often the case in ML). Even if an algorithm performs admirably on one multicore system, once we make the transition to a cluster of machines, we find the limitations of distributed environments quickly brought to fore:

1. Network communication is slow, with higher latency and lower bandwidth than intra-machine communication, so computational threads must spend more time communicating with each other. Furthermore, ML algorithms greatly stress network communication limits because they often have large shared parameters or intermediate state, with variables that are comparable in size to the input data (or even larger), and which must be read by all computational threads. Storing and communicating these variables is challenging in a distributed environment.

2. No shared address space. Threads on different machines cannot read each other's variables, necessitating other styles of communication.

3. Synchronization is expensive. As a consequence of the above two points, getting threads on different machines to "see" the same values takes much more time (and is more tedious to program) than on a single machine.

While such limitations pose less of an issue with data-parallel algorithms, the majority of ML algorithms are not data-parallel. Rather, they possess *global* shared state or intermediate values that must be available to all computational threads. Thus effort and time must be expended to synchronize these values across different machines.

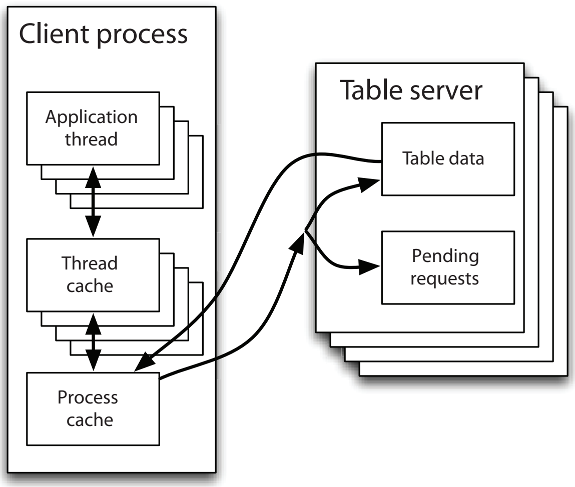


Figure 1. LazyTables System Diagram.

The task, then, is to communicate and synchronize global state effectively across a distributed cluster of machines, preferably without overtaxing the limited communication bandwidth made available by the cluster's network. Unfortunately,

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 Masters program for "Big Data" systems. Along the way, many students graduated and joined PDL Consortium companies, new students joined PDL, and many cool papers have been published. Let me highlight a few things.

It's exhilarating to be a researcher whose topic-space is at the core of a major growth area (and source of hype)… and PDL finds itself at the core of two of them: cloud computing and Big Data. Of course, we have been working in both areas since long before either buzzword was coined, but the opportunities for impact have grown with the hype. Last year, we took a leadership role in the establishment of the Intel Science and Technology Center for Cloud Computing (ISTC-CC), which was described more fully in last year's PDL Packet; our leadership role in that 5-institution activity focused on cloud computing infrastructure continues. We also continue "Big Data" systems research, and we have started a couple of cool new efforts there too.

One significant new development has been the establishment of a new Masters program that trains students with strong practical skills in the creation and exploitation of systems for Big Data analytics. Garth has been the primary driving force behind this program, and it is described in some detail in a short article in this PDL Packet. Among other interesting aspects, PDL Consortium companies may want to take advantage of the opportunity to host such students on 7-month internships that can satisfy the program's capstone project requirement.

A fair number of us have also started collaborating closely with some of Carnegie Mellon's excellent machine learning faculty to explore new programming systems for Big Data activities. While the Map-Reduce approach is good for very simple data processing tasks, it is a very poor tool for many of the advanced machine learning techniques that give "Big Data" its great promise. Such algorithms involve the types of data sharing and iterative work that simply don't fit Map-Reduce's strict data-parallel model. The front-page article describes one of the new approaches that have come from brainstorming and working with the machine learning folks, and the ideas are flowing fast and furious in this cross-domain collaboration. I expect lots of cool stuff in this area, going forward.

Of course, we also continue several of our other activities in the general space of data-intensive computing. For example, we continue to operate a DISC service for CMU researchers, based on the Hadoop software stack, and pursue our efforts to converge cloud databases and huge-scale parallel file systems—the two are separately evolving toward similar solutions, and we envision high-level frameworks and mechanisms that work well for both. Examples include scalable metadata support, such as GIGA+, and cool approaches to high-ingest updates for metadata services and cloud databases generally.

Although I'm talking about it less, here, we continue to explore a lot of cool new approaches to cloud computing infrastructure as well. One active area involves making elastic storage systems and exploiting their nature for resource efficiency, controlled resource sharing, and avoiding efficiency-reducing interference among different applications. Another active area seeks to generalize the FAWN concept

THE PDL PACKET

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

PUBLISHER
Greg Ganger

EDITOR
Joan Digney

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

THE PDL LOGO

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

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

PARALLEL DATA LABORATORY

FACULTY

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

Anastasia Ailamaki
David Andersen
Lujo Bauer
Chuck Cranor
Lorrie Cranor
Christos Faloutsos
Eugene Fink
Rajeev Gandhi
Garth Gibson
Seth Copen Goldstein
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
Charlene Zang

VISITING RESEARCHERS / POST DOCS

Samira Khan
Hiroshi Mine

GRADUATE STUDENTS

Pavan Kumar Alampalli
Rachata Ausavarungnirun
Ben Blum
Jim Cipar
Henggang Cui
Alex Degtiar
Bin Fan
Bin Fu
Anshul Gandhi
Ge Gao
Samantha Gottlieb
Fan Guo
Wesley Jin
Chinmay Kamat
Mike Kasick
Soila Pertet Kavulya
Peter Klemperer
Elie Krevat
Hyeontaek Lim
Adit Madan
Michelle Mazurek
Justin Meza
Iulian Moraru
Luca Parolini
Swapnil Patil
Praveen Ramakrishnan
Kai Ren
Wolfgang Richter
Raja Sambasivan
Vivek Seshadri
Ilari Shafer
Jiri Simsa
Yu Su
Jiaqi Tan
Wittawat Tantisiriroj
Alexey Tumanov
Yifan Wang
Jinliang Wei
Lin Xiao
Lianghong Xu
HanBin Yoon
Chaomin Yu
Timothy Zhu

FROM THE DIRECTOR'S CHAIR

of specializing the hardware platform to the application in order to create cloud computing infrastructures composed of heterogeneous mixes of systems, fully exploited to maximize efficiency while also benefiting from economies of scale.

We continue to explore ways of exploiting the exciting new underlying storage technologies, such as NVM and Flash SSDs, to improve systems. Even the disk drive is changing, with technologies like shingled magnetic recording creating a need to reconsider usage patterns and interfaces. The lower cost-per-bit and energy-per-bit nature of NVM, but higher latencies, creates a compelling case for hybrid main memories (together with DRAM). It is also interesting to revisit architectures like "active disks" in the context of Flash-based SSDs, given their internal parallelism and bandwidth characteristics.

Another of our long-standing focuses, automation, especially of problem diagnosis, continues aggressively to seek answers to making systems like distributed storage and Big Data systems more robust and easier to manage. It is clear that there will be no silver bullet here, and PDL research is probing a number of complementary paths. In fact, several of our new approaches have really matured nicely, resulting in multiple PhD students defending cool theses (and graduating) this year. The abstracts for their dissertations, found later in this PDL Packet, describe them better than I could here.

Many other ongoing PDL projects are also producing cool results. For example, our FAWN research continues to explore how new technologies (e.g., Flash and NVM) can be exploited in specialized system architectures for important classes of applications (e.g., key-value stores and Big Data systems). The FAWN team also continues to win Joulesort competitions by applying their insights. Our continued operation of private clouds in the Data Center Observatory (DCO) serves the dual purposes of providing resources for real users (CMU researchers) and providing us with invaluable Hadoop logs, instrumentation data, and case studies. The latter data from these systems has been invaluable in the problem diagnosis research discussed above, as well as other work on Big Data tools, cluster resource scheduling, and elastic storage policies. 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.

A birds-eye view of one of the Retreat 2012 poster sessions at Bedford Springs.

YEAR IN REVIEW

May 2013

- 15th Annual PDL Spring Industry Visit Day.
- Raja Sambasivan defended his dissertation "Diagnosing Performance Changes in Distributed Systems by Comparing Request Flows."
- Bin Fu defended his dissertation "Algorithms for Large-Scale Astronomical Problems."
- Soila Pertet Kavulya defended her dissertation on "Statistical Diagnosis of Chronic Problems in Production Systems."
- Ben Blum will be interning with Mozilla this summer.
- Pavan Kumar Alampalli will be employed by Google at their Mountain View Campus following his graduation.
- Praveen Kumar Ramakrishnan will be working with Facebook following his graduation this spring.

April 2013

- Swapnil V. Patil defended his dissertation "Scale and Concurrency of Massive File System Directories."

March 2013

- Wolf Richter was awarded an IBM Ph.D. fellowship.

February 2013

- Iulian Moraru proposed his Ph.D. thesis research on "More Consensus with Egalitarian Parliaments."
- PDL received Intel funding for cloud computing research.
- David Andersen, with Michael Kaminsky (Intel), received the Allen Newell Award for Research Excellence.

January 2013

- Onur Mutlu has been appointed to the Dr. William D. and Nancy W. Stecker Early Career Professor chair in ECE.

December 2012

- Soila Pertet Kavulya and co-authors Elmer Garduno, Jiaqi Tan, Rajeev Gandhi, and Priya Narasimhan received the Best Student Paper Award at USENIX LISA 2012 for "Theia: Visual Signatures for Problem Diagnosis in Large Hadoop Clusters."
- Michelle Mazurek proposed her Ph.D. research on "A Tag-Based, Logical Access-Control Framework for Personal Data Sharing."
- Bin Fan proposed his Ph.D. thesis research: "Building Memory-efficient Key-Value Clusters with Provable Bounds Using Better Hashing and Selected Caching."
- Garth Gibson was named an ACM Fellow.

November 2012

- 20th Annual PDL Retreat!
- Onur Mutlu won an Intel Early Career Award for his innovative research.
- The Future of Privacy Forum has listed two of Lorrie Cranor's group's papers as 2012 Privacy Policy Makers.
- Garth Gibson was keynote speaker at the Storage System, Hard Disk and Solid State Technologies Summit in Singapore.

October 2012

- Jim Cipar proposed his dissertation research on "Trading Latency for Freshness in Storage Systems."
- Vijay Vasudevan's dissertation "Energy-efficient Data-intensive Computing with a Fast Array of Wimpy Nodes" tied for top SCS dissertation. Duen Horng "Polo" Chau's research on "Data Mining Meets HCI: Making Sense of Large Graphs" received honorable mention.
- Christos Faloutsos and his team won a National Science Foundation Big Data Award.

September 2012

- Amar Phanishayee defended his dissertation "Chaining for Flexible and High-Performance Key-Value Systems."
- B. Aditya Prakash defended his dissertation "Understanding and Managing Propagation on Large Networks—Theory, Algorithms, and Models."
- Garth Gibson was keynote speaker at the 2012 Storage Developer Conference in Santa Clara, CA.

August 2012

- Leman Akoglu defended her dissertation "Mining and Modeling Real-world Networks: Patterns, Anomalies, and Tools."

July 2012

- PDL Alum Lei Li's research on "Fast Algorithms for Mining Co-evolving Time Series" was the runner-up for the 2012 Doctoral Dissertation Award from ACM's SIG in Data Mining.

May 2012

- Ben Blum presented his M.S. thesis research "Landslide: Systematic Dynamic Race Detection in Kernel-space."
- Christos Faloutsos received an honorary degree from Aristotle University.
- PDL Alum Ryan Johnson won the SIGMOD Jim Gray Doctoral Dissertation Award.
- The PDL/ISTC-CC team won 3 categories in the 2012 JouleSort competition.

April 2012

- Wolf Richter received the Alan J. Perlis SCS Student Teaching Award.
- 14th Annual PDL Spring Industry Visit Day.

Asymmetry-aware Execution Placement on Manycore Chips

Tumanov, Wise, Mutlu & Ganger

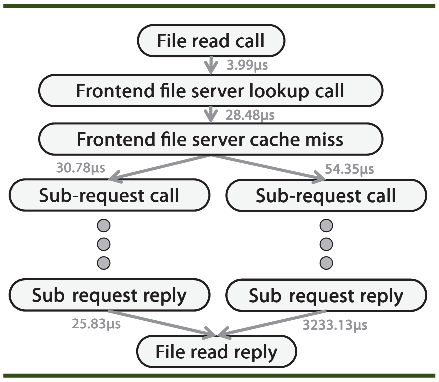
3rd Workshop on Systems for Future Multicore Architectures (SFMA'13), EuroSys'13, April 14-17, 2013, Prague, Czech Republic.

Network-on-chip based manycore systems with multiple memory controllers on a chip are gaining prevalence. Among other research considerations, placing an increasing number of cores on a chip creates a type of resource access asymmetries that didn't exist before. A common assumption of uniform or hierarchical memory controller access no longer holds. In this paper, we report on our experience with memory access asymmetries in a real manycore processor, the implications and extent of the problem they pose, and one potential thread placement solution that mitigates them. Our user-space scheduler harvests memory controller usage information generated in kernel space on a per process basis and enables thread placement decisions informed by threads' historical physical memory usage patterns. Results reveal a clear need for low-overhead, per-process memory controller hardware counters and show improved benchmark and application performance with a memory controller usage-aware execution placement policy.

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

Sambasivan, Shafer, Mazurek & Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-13-104 (supersedes CMU-PDL-12-102), April 2013.



Example request-flow graph showing the flow of a read request through a distributed storage system. Node names represent important events observed on the various components while completing the required work. Edges show latencies between these events. Fan-outs represent the start of parallel activity, and synchronization points are indicated by fan-ins. Due to space constraints, only the events observed on the frontend file server are shown.

Distributed systems are complex to develop and administer, and performance problem diagnosis is particularly challenging. When performance degrades, the problem might be in any of the system's many components or could be a result of poor interactions among them. Recent research efforts have created tools that automatically localize the problem to a small number of potential culprits, but effective visualizations are needed to help developers understand and explore their results. This paper compares side-by-side, diff, and animation-based approaches for visualizing the results of one proven automated localization technique called request-flow comparison. Via a 26-person user study, which included real distributed systems developers, we identify the unique benefits that each approach provides for different usage modes and problem types.

The Impact of Length and Mathematical Operators on the Usability and Security of System-assigned One-time PINs

Kelley, Komanduri, Mazurek, Shay, Vidas, Bauer, Christin & L.Cranor

2013 Workshop on Usable Security (USEC), April 2013.

Over the last decade, several proposals have been made to replace the common personal identification number, or PIN, with often-complicated but theoretically more secure systems. We present a case study of one such system, a specific implementation of system-assigned one-time PINs called PassGrids. We apply various modifications to the basic scheme, allowing us to review usability vs. security tradeoffs as a function of the complexity of the authentication scheme. Our results show that most variations of this one-time PIN system are more enjoyable and no more difficult than PINs, although accuracy suffers for the more complicated variants. Some variants increase resilience against observation attacks, but the number of users who write down or otherwise store their password increases with the complexity of the scheme. Our results shed light on the extent to which users are able and willing to tolerate complications to authentication schemes, and provides useful insights for designers of new password schemes.

Solving the Straggler Problem with Bounded Staleness

Cipar, Ho, J. Kim, Lee, Ganger, Gibson, Keeton & Xing

14th USENIX HotOS Workshop, Santa Ana Pueblo, NM, May 13-15, 2013.

Many important applications fall into the broad class of iterative convergent algorithms. Parallel implementation of these algorithms are naturally expressed using the Bulk Synchronous Parallel (BSP) model of computation. However, implementations using BSP are plagued by the straggler problem, where every transient slowdown of any given thread can delay all other threads. This paper presents the Stale Synchronous Parallel (SSP) model as a generalization of BSP that preserves many of its advantages, while avoiding the straggler problem. Algorithms using SSP can execute efficiently, even with significant delays in some threads, addressing the oft-faced straggler problem.

continued on page 6

continued from page 5

MISE: Providing Performance Predictability and Improving Fairness in Shared Main Memory Systems

Subramanian, Seshadri, Y. Kim, Jaiyen & Mutlu

19th International Symposium on High-Performance Computer Architecture (HPCA 2013), Shenzhen, China, February 2013.

Applications running concurrently on a multicore system interfere with each other at the main memory. This interference can slow down different applications differently. Accurately estimating the slowdown of each application in such a system can enable mechanisms that can enforce quality-of-service. While much prior work has focused on mitigating the performance degradation due to inter-application interference, there is little work on estimating slowdown of individual applications in a multi-programmed environment. Our goal in this work is to build such an estimation scheme.

To this end, we present our simple Memory-Interference-induced Slowdown Estimation (MISE) model that estimates slowdowns caused by memory interference. We build our model based on two observations. First, the performance of a memory-bound application is roughly proportional to the rate at which its memory requests are served, suggesting that request-service-rate can be used as a proxy for performance. Second, when an application's requests are prioritized over all other applications' requests, the application experiences very little interference from other applications. This provides a means for estimating the uninterfered request-service-rate of an application while it is run alongside other applications. Using the above observations, our model estimates the slowdown of an application as the ratio of its uninterfered and interfered request service rates. We propose simple changes to the above model to estimate the slowdown of non-memory-bound applications.

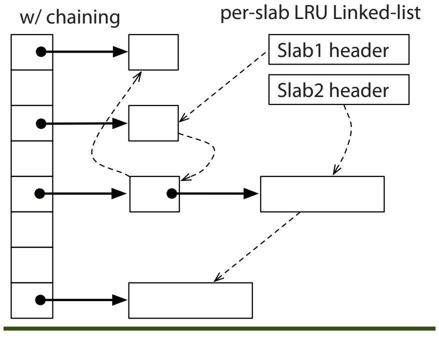
We demonstrate the effectiveness of our model by developing two new memory scheduling schemes: 1) one that provides soft quality-of-service guarantees and 2) another that explicitly attempts to minimize maximum slowdown (i.e., unfairness) in the system. Evaluations show that our techniques perform significantly better than state-of-the-art memory scheduling approaches to address the above problems.

MemC3: Compact and Concurrent Memcache with Dumber Caching and Smarter Hashing

Fan, Andersen & Kaminsky

10th USENIX NSDI, Apr 2013. Supersedes Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-116. November 2012.

This paper presents a set of architecturally and workload-inspired algorithmic and engineering improvements to the popular Memcached system that substantially improve both its memory efficiency and throughput. These techniques—optimistic cuckoo hashing, a compact LRU-approximating eviction algorithm based upon CLOCK, and comprehensive implementation of optimistic locking—enable the resulting system to use 30% less memory for small key-value pairs, and serve up to 3x as many queries per second over the network. We have implemented these modifications in a system we call MemC3—Memcached with CLOCK and Concurrent Cuckoo hashing—but believe that they also apply more generally to many of today's read-intensive, highly concurrent networked storage and caching systems.



Memcached data structures.

Application-to-Core Mapping Policies to Reduce Memory System Interference in Multi-Core Systems

Das, Ausavarungnirun, Mutlu, Kumar & Azimi

19th International Symposium on High-Performance Computer Architecture (HPCA 2013), Shenzhen, China, February 2013.

Future many-core processors are likely to concurrently execute a large number of diverse applications. How these applications are mapped to cores largely determines the interference between these applications in critical shared hardware resources. This paper proposes new application-to-core mapping policies to improve system performance by reducing interapplication interference in the on-chip network and memory controllers. The major new ideas of our policies are to: 1) map network-latency-sensitive applications to separate parts of the network from network-bandwidth-intensive applications such that the former can make fast progress without heavy interference from the latter, 2) map those applications that benefit more from being closer to the memory controllers close to these resources.

Our evaluations show that, averaged over 128 multiprogrammed workloads of 35 different benchmarks running on a 64-core system, our final application-to-core mapping policy improves system throughput by 16.7% over a state-of-the-art baseline, while also reducing system unfairness by 22.4% and average interconnect power consumption by 52.3%.

continued on page 7

continued from page 6

So, You Want to Trace Your Distributed System? Key insights from years of practical experience

Sambasivan, Fonseca, Shafer & Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-13-103, February 2013.

End-to-end tracing captures the workflow of servicing individual requests within and across components of a distributed system. As distributed systems grow in scale and complexity, such tracing is becoming a critical tool for management tasks like diagnosis and resource accounting. Drawing upon our experiences building and using end-to-end tracing infrastructures and previous research, this paper distills design axes that determine trace utility for important use cases. Developers should explicitly consider design choices for these axes, lest their tracing infrastructure fail to satisfy a sufficient range of uses. We identify good design choices, contrast them to choices made by previous implementations, and show where prior implementations fall short. We also identify remaining challenges on the path to making tracing an integral part of distributed system design.

Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture

Lee, Y. Kim, Seshadri, Liu, Subramanian & Mutlu

19th International Symposium on High-Performance Computer Architecture (HPCA), Shenzhen China, February 2013.

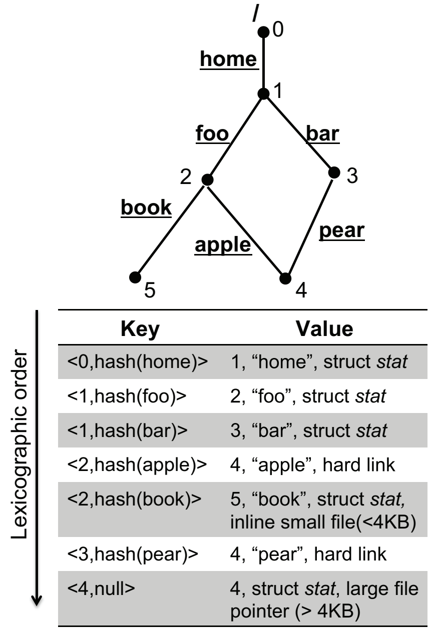
The capacity and cost-per-bit of DRAM have historically scaled to satisfy the needs of increasingly large and complex computer systems. However, DRAM latency has remained almost constant, making memory latency the performance bottleneck in today's systems. We observe that the high access latency is not intrinsic to DRAM, but a trade-off made to decrease cost-per-bit. To mitigate the high area overhead of DRAM sensing structures, commodity DRAMs connect many DRAM cells to each sense-amplifier through a wire called a bitline. These bitlines have a high parasitic capacitance due to their long length, and this bitline capacitance is the dominant source of DRAM latency. Specialized low-latency DRAMs use shorter bitlines with fewer cells, but have a higher cost-per-bit due to greater senseamplifier area overhead. In this work, we introduce Tiered- Latency DRAM (TL-DRAM), which achieves both low latency and low cost-per-bit. In TL-DRAM, each long bitline is split into two shorter segments by an isolation transistor, allowing one segment to be accessed with the latency of a short-bitline DRAM without incurring high cost-per-bit. We propose mechanisms that use the low-latency segment as a hardware-managed or software-managed cache. Evaluations show that our proposed mechanisms improve both performance and energy-efficiency for both single-core and multi-programmed workloads.

TABLEFS: Enhancing Metadata Efficiency in the Local File System

Ren & Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-13-102, January 2013. Revised version of CMU-PDL-12-110.

File systems that manage magnetic disks have long recognized the importance of sequential allocation and large transfer sizes for file data. Fast random access has dominated metadata lookup data structures with increasing use of B-trees on-disk. Yet our experiments with workloads dominated by metadata and small file access indicate that even sophisticated local disk file systems like Ext4, XFS and Btrfs leave a lot of opportunity for performance improvement in workloads dominated by metadata and small files.



Key	Value
<0,hash(home)>	1, "home", struct *stat*
<1,hash(foo)>	2, "foo", struct *stat*
<1,hash(bar)>	3, "bar", struct *stat*
<2,hash(apple)>	4, "apple", hard link
<2,hash(book)>	5, "book", struct *stat*, inline small file(<4KB)
<3,hash(pear)>	4, "pear", hard link
<4,null>	4, struct *stat*, large file pointer (> 4KB)

An example illustrates table schema used by TABLEFS's metadata store. The file with inode number 4 has two hard links, one called "apple" from directory foo and the other called "pear" from directory bar.

In this paper we present a stacked file system, TABLEFS, which uses another local file system as an object store. TABLEFS organizes all metadata into a single sparse table backed on disk using a Log-Structured Merge (LSM) tree, LevelDB in our experiments. By stacking, TABLEFS asks only for efficient large file allocation and access from the local file system. By using an LSM tree, TABLEFS ensures metadata is written to disk in large, non-overwrite, sorted and indexed logs. Even an inefficient FUSE based user level implementation of TABLEFS can perform comparably to Ext4, XFS and Btrfs on data-intensive benchmarks, and can outperform them by 50% to as much as 1000% for metadata-intensive workloads. Such promising performance results from TABLEFS suggest that local disk file systems can be significantly improved by more aggressive aggregation and batching of metadata updates.

continued on page 19

AWARDS & OTHER PDL NEWS

March 2013

Wolfgang Richter Awarded IBM Ph.D. Fellowship

Congratulations to Wolf Richter, who will be receiving an IBM Ph.D. Fellowship. The IBM Ph.D. Fellowship Awards Program is a worldwide program, which honors exceptional Ph.D. students who have an interest in solving problems of interest to IBM and which are fundamental to innovation including, innovative software, new types of computers, technology, and interdisciplinary projects that create social and business value. The 2013-2014 Fellowship begins in the fall semester of 2013 and covers the academic year; an associated internship may be a summer assignment in 2013 or 2014.

March 2013

Welcome Domenico!

The Sinopolis are proud to announce the arrival of their son Domenico Emanuele, who was born at 12.01PM on March 15, 2013. His Dad assures us Domenico is a proud Pittsburgher!

March 2013

Welcome Larkin!

Mitch and Stephanie are thrilled to present their new daughter Larkin Dolores Franzos, who was born on Sunday March 10, at 0:38 EST, at 6 lbs. 13 oz and 20.75 inches. Congratulations!

February 2013

PDL Receives Intel Funding for Cloud Computing Research

Carnegie Mellon University's Parallel Data Lab (PDL) has received funding for cloud computing research from Intel, the world's largest manufacturer of semiconductor products.

"This financial support affords us an excellent platform for open collaboration research into the underlying technologies so essential to allowing cloud computing to reach the promise of dramatically improving efficiency, ubiquity and productivity for all scales of user-facing applications across many areas of information technology," said PDL Director Gregory Ganger, the Stephen F. Jatras Professor of Electrical and Computer Engineering at CMU. Ganger is also a co-principal investigator of the Intel Science and Technology Center (ISTC) for cloud computing at CMU along with Phil Gibbons, an Intel research scientist and an adjunct professor in computer science.

"This support helps drive development and implementation of strategies to explore emerging technologies within a university research environment," said Scott Buck, university program officer for Intel.

Ganger, an expert in the risks and benefits of cloud computing, reports that cloud computing has the potential to provide large efficiency improvements for both industry and federal government information technology functions. Cloud computing involves using someone else's computers (and possibly software) to accomplish a task rather than one's own. Ganger has recommended that the U.S. government support standardization and research experimentation efforts in pursuit of cloud computing's potential.

For more than a decade, Intel has supported novel research by CMU faculty, including research into embedded computing designed to transform future experiences in the home, car and retail environment.

-- ECE News, Feb. 25, 2013

February 2013

ISTC-CC Researchers Awarded Allen Newell Award for Research Excellence!

Congratulations to David Andersen and Michael Kaminsky, recipients of the Allen Newell Award for Research Excellence!

The Allen Newell Award for Research Excellence recognizes an outstanding body of work that epitomizes Allen Newell's research style as expressed in his words: "Good science responds to real phenomena or real problems. Good science is in the details. Good science makes a difference."

David and Michael won the award for "Energy-efficient Data Intensive Computing" for their FAWN project. FAWN (Fast Array of Wimpy Nodes) demonstrates how many low-power (e.g., Atom) nodes with SSDs provides significant energy-efficiency gains for important cloud workloads such as key-value stores, and how to redesign system software to get the maximum benefit from such platforms.

January 2013

Onur Mutlu Appointed to Chair

Congratulations to Onur Mutlu who has been appointed as the "Dr. William D. and Nancy W. Stecker Early Career Professor" in ECE, effective January 1, 2013.

continued on page 9

continued from page 8

December 2012

PDL Group Wins Best Student Paper Award at USENIX LISA 2012

Congratulations to Soila Pertet Kavulya and co-authors Elmer Garduno, Jiaqi Tan, Rajeev Gandhi, and Priya Narasimhan, for winning Best Student Paper at 26th USENIX Large Installation System Administration Conference (LISA'12), Dec 9-14, San Diego, CA for their work on visualization for failure diagnosis in the paper "Theia: Visual Signatures for Problem Diagnosis in Large Hadoop Clusters."

December 2012

Garth Gibson Named ACM Fellow

Congratulations to Garth, who has been named a Class of 2012 ACM Fellow for "contributions to the performance and reliability of storage systems." CMU PDL alum Ion Stoica was also named. The full list of awardees can be found here: http://www.acm.org/press-room/news-releases/2012/fellows-2012.

November 2012

Onur Mutlu Wins Intel Early Career Award For Innovative Research

Carnegie Mellon University's Onur Mutlu has received the prestigious 2012 Intel Early Career Faculty Award for outstanding research and educational contributions in the field of computer architecture.

Intel's Early Career Faculty Honor Program award provides financial and networking support to those faculty members who are early in their careers and who show great promise as future academic leaders in disruptive computing technologies. The purpose of the program is to help promote the careers of promising early career faculty members and to foster long-term collaborative relationships with senior technical leaders at Intel. The $40,000 award is designed to cover some research costs and travel.

A large focus of Mutlu's current research is on new memory architecture and technologies to make computers store and manipulate data more efficiently and reliably.

Mutlu's research has received several other prestigious recognitions, including numerous best paper awards and "Top Pick" paper selections by the Institute of Electrical and Electronics Engineers (IEEE) Micro journal. In 2011, he received the Young Computer Architect Award from IEEE Computer Society's Technical Committee on Computer Architecture. And in 2012, CMU's College of Engineering recognized him with the George Tallman Ladd Research Award.

-- expanded article in CMU News, Nov. 28, 2012

November 2012

Lorrie Cranor's Papers Lead in Privacy Writings

The Future of Privacy Forum (FPF) has selected the 2012 Privacy Papers for Policy Makers, highlighting eight leading privacy writings that were voted by the FPF Advisory Board to be most useful for policy makers. The selected writings include two papers authored by Carnegie Mellon faculty and Ph.D. students. They are: "Smart, Useful, Scary, Creepy: Perceptions of Online Behavioral Advertising," by Blase Ur (Ph.D. candidate, Institute for Software Research), Pedro G. Leon (Ph.D. candidate, Engineering and Public Policy), Lorrie Faith Cranor (associate professor, ISR and EPP), Richard Shay (Ph.D. candidate, ISR) and Yang Wang (former post-doc at CyLab).

Earning notable mention was "Why Johnny Can't Opt Out: A Usability Evaluation of Tools to Limit Online Behavioral Advertising," by Pedro G. Leon, Blase Ur, Rebecca Balebako (Ph.D. candidate, EPP), Lorrie Faith Cranor, Richard Shay and Yang Wang.

Read more at http://www.futureofprivacy.org/privacy-papers-2012/

-- CMU 8.5x11 News, Nov 1, 2012

November 2012

Garth Gibson Keynote Speaker

Garth was an invited keynote speaker at two conferences recently. His talk on "Storage Systems Issues for Shingled Magnetic Recording" opened the Storage System, Hard Disk and Solid State Technologies Summit, co-located with the Asia-Pacific Magnetic Recording Conference (APMRC), in Singapore, November 1, 2012. In September, he gave the SNIA SDC Keynote talk "Storage Systems for Shingled Disks" at the 2012 Storage Developer Conference in Santa Clara, CA.

October 2012

SCS Dissertation Award Winners Announced

Congratulations to the following PDL award winners! "Energy-efficient Data-intensive Computing with a Fast Array of Wimpy Nodes" by Vijay Vasudevan (Advisor: David Andersen) was one of two dissertations chosen for the top award. Receiving Honorable Mention is Duen Horng "Polo" Chau (Advisor: Christos Faloutsos) and his research on "Data Mining Meets HCI: Making Sense of Large Graphs."

continued on page 10

continued from page 9

October 2012

CMU Repurposing Supercomputers From Los Alamos National Lab

The National Science Foundation, the New Mexico Consortium and Carnegie Mellon University joined forces to launch PRObE, a one-of-a-kind supercomputer research center using a cluster of 2,048 recently retired computers. The Tribune-Review reports, "although the main facility will remain in Los Alamos, Carnegie Mellon's Parallel Data Lab in Pittsburgh will house two smaller centers." Garth Gibson, who collaborated on the project, described the Pittsburgh facility as a 'staging cluster,' which will allow researchers to perform small experiments and demonstrate to the PRObE committee that they're ready to request time on the facility in Los Alamos."

--info from the Pittsburgh Tribune-Review, Oct. 23, 2012

October 2012

Christos Faloutsos and Team Win Big Data Award

The National Science Foundation (NSF), with support from the National Institutes of Health (NIH), recently announced nearly $15 million in new Big Data fundamental research projects. These awards aim to develop new tools and methods to extract and use knowledge from collections of large data sets to accelerate progress in science and engineering research and innovation.

Christos Faloutsos (PI), Tom Mitchell (co-PI) and their team proposed a successful project "BIGDATA: Mid-Scale: DA: Collaborative Research: Big Tensor Mining: Theory, Scalable Algorithms and Applications." The objective of the project is to develop theory and algorithms to tackle the complexity of language processing, and to develop methods that approximate how the human brain works in processing language. The research also promises better algorithms for search engines, new approaches to understanding brain activity, and better recommendation systems for retailers.

--from NSF Press Release 12-187

July 2012

PDL Alum Receives ACM SIGKDD Dissertation Honors

Lei Li, now a post-doc at the University of California, Berkeley, (SCS 2011), was the runner up for the prestigious 2012 Doctoral Dissertation Award from the Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining (ACM SIGKDD).

In his dissertation, "Fast Algorithms for Mining Co-evolving Time Series," Li developed novel algorithms for forecasting, clustering and missing-value imputation for time sequences in a broad spectrum of settings, from motion-capture sequences to data-center monitoring.

The ACM SIGKDD dissertation award is the highest distinction for a PhD in the field.

--from CMU News, July 27, 2012

May 2012

Faloutsos to Receive Honorary Degree from Aristotle University

The Aristotle University of Thessaloniki, the largest university in Greece, will award an honorary doctorate degree to Christos Faloutsos, professor of computer science. The title of Doctor Honoris Causa will be conferred to Faloutsos during a May 30 convocation. During convocation, Faloutsos will present a convocation address on "Mining Large Social Networks: Patterns and Anomalies."

Faloutsos' research interests include data mining for graphs and streams, fractals, database performance and indexing for multimedia and bio-informatics data, and his cross-disciplinary work is widely and regularly cited.

--from CMU News May 30, 2012

May 2012

PDL Alum Ryan Johnson Wins SIGMOD Jim Gray Doctoral Dissertation Award

Congratulations to Ryan Johnson, who has received the very prestigious SIGMOD Jim Gray Doctoral Dissertation Award for his PhD thesis titled "Scalable Storage Managers for the Multicore Era"! SIGMOD has established the award to recognize excellent research by doctoral candidates in the database field. This award, which was previously known as the SIGMOD Doctoral Dissertation Award, was renamed in 2008 with the unanimous approval of ACM Council in honor of Dr. Jim Gray.

May 2012

David Andersen Collaborates with Intel ISTC-CC Researchers to Win JouleSort Competition!

An ISTC-CC team—Babu Pillai, Michael Kaminsky, Mike Kozuch (Intel), and Dave Andersen (CMU)—were announced winners in 3 categories of the 2012 JouleSort competition, setting new records for fewest joules needed to sort 108, 109, and 1010 records. The team used an Intel Core i7-2700K desktop processor, coupled with 16 Intel 710 Series SSDs to beat existing energy efficiency records in the 10GB, 100GB, and 1TB categories.

April 2012

Wolf Richter Receives Alan J. Perlis SCS Student Teaching Award

Congratulations to Wolfgang Richter, who has received the Alan J. Perlis Graduate Student Teaching Award for 2012. The awards, for both graduate and undergraduate teaching assistants, are based on student nominations, recommendation letters and reviews, and honors the students who have shown the highest degree of excellence and dedication as teaching assistants.

NEW PDL-DESIGNED MASTERS IN "BIG DATA" SYSTEMS

Garth Gibson

In the last eight months PDL's Garth Gibson has been guiding the first class enrolled in a new masters track for "Big Data" Systems. The new track is part of the 10 year old Very Large Information Systems (VLIS) masters program. Primarily taught by PDL faculty, the new track emphasizes computer science courses such as distributed systems, storage systems, cloud computing, data mining, parallel computer architecture, and parallel programming. The parent VLIS program also offers courses in machine learning analytics, natural language processing, database applications, and software engineering. For the students taking this new track the goal is to "drink from the fire hose" a state-of-the-art one-year program of project-oriented systems courses, spend a final semester primarily doing a hands-on capstone research project with a PDL-class advisor, and graduate after 16 months with an exciting job at a high-tech company soon to earn a six figure salary. For PDL, its sponsors and partners, the goal is to increase the annual number of solidly trained engineers graduating with a masters degree from CMU.

The first class in the systems track contains five students; the second class has been selected and will contain 18 students. We'll see next year where class size goes in the future.

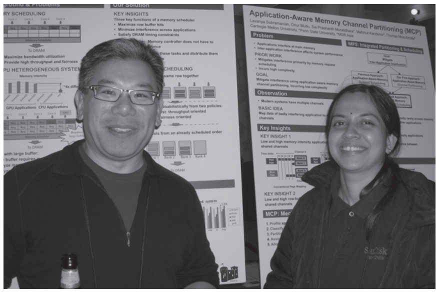
The first class will soon finish their course work. On average this class obtained a 3.8 GPA in their first semester courses, while the overall average in these classes was considerably lower. Based on the mid-semester estimated grades in the current semester, they are achieving a 3.7 GPA. This puts these students in the top quadrant of CMU graduates, and promises a strong standard of academic accomplishment. Of the eight courses they will soon finish, at least five have to be selected from the eight systems track core courses. At a minimum, three of these must be at least 40% based on systems projects. Examples of projects include implementing a quorum consensus replicated state machine protocol, incremental log processing with MapReduce and a NoSQL database, a laptop storage system splitting files between the local disk and a deduplicated cloud storage service, and a scalable metadata service for HDFS balanced over many independent servers. For electives, students in this program typically study machine learning or software engineering.

The first class is about to start their summer internships. Two students elected to spend 12 weeks at Amazon and two chose to spend 12 weeks at Google. These internship positions were not arranged by CMU program staff. Instead, the students took advantage of the general internship services offered by the various technical opportunities job fairs, some additional advertising by VLIS leadership and, of course, the PDL jobs list. These internships offer practical training for the students, but are not otherwise managed by VLIS leadership. The students will return to CMU in the fall and undertake a full-semester capstone project in which their skills will be employed to create a significant R&D activity. They will graduate in December with an MS-IT from the Systems Track of the Very Large Information Systems program. Most will probably accept a full time job in the industry some time during the fall.

The fifth student, however, is the first to pursue a different option for the second half of his masters program. An internship sponsoring company, IBM, and a supervisor at that company have agreed to his proposal to participate in a seven-month-long combined internship and capstone project located at the company's facilities. He will graduate with the same degree at the same time as the other students because the capstone project will have been done in parallel with this long internship. For a student, the idea is to find an industry (or government) supervisor with expertise and hands-on experience not necessarily available from CMU and add to their resume an unusual, substantial and differentiated project. For CMU, the idea is to strengthen our ties with our customers and sponsors, link our students to the rapidly changing technology landscape as soon as possible, and use this to differentiate our education from our competitors. For the internship sponsors, this allows a lengthy internship significantly improving hiring decisions and recruiting effectiveness.

The seven-month combined internship and capstone project option is defined by the student and internship sponsor, approved and monitored by CMU and then evaluated by CMU with input from an internship supervisor. It is designed to make it possible for the student to carry a full time internship that does not contribute to the project that CMU monitors and evaluates. However, this sort of strongly partitioned plan is not favored because a collaborative capstone project between the student and the internship sponsor could be much more effective. We hope that the internship supervisor will see a way for a piece of the internship to be an appropriate capstone project, allowing enough to be monitored and reported to CMU without an NDA. This could lead to strong engagement from the supervisor and from CMU, ensuring that combined internship and capstone projects are seen to be more valuable than two separate activities.

If this seven month combined internship and capstone project interests you, or worries you, seek Garth out and help him shape this program to be more successful for all.



Brian Hirano, of Oracle and Lavanya Subramanian discuss her research on "Application-Aware Memory Channel Partitioning."

DISSERTATION ABSTRACT: Diagnosing Performance Changes in Distributed Systems by Comparing Request Flows

Raja Sambasivan

Carnegie Mellon University ECE Ph.D. Dissertation, CMU-PDL-13-105, May 6, 2013

Diagnosing performance problems in modern datacenters and distributed systems is incredibly challenging, as the root cause could be contained in any one of the system's numerous components or subcomponents, or worse, could be a result of interactions among them. As distributed systems continue to increase in complexity, diagnosis tasks will only become more challenging. Clearly, there is an urgent need for a new class of diagnosis techniques capable of helping developers fix problems in distributed environments.

As a step toward addressing this need, this dissertation proposes a novel technique, called request-flow comparison, for automatically localizing the sources of performance changes from the myriad potential culprits in the distributed system to just a few potential ones. Request-flow comparison works by contrasting the workflow of how individual requests are serviced within and among every component of the distributed system between two periods: a non-problem period and a problem period. By identifying and ranking performance-affecting changes, request-flow comparison provides developers with promising starting points for their diagnosis efforts. Request workflows are obtained with less than 1% overhead via use of recently developed end-to-end tracing techniques.

PDL alum, Hugo Patterson, listens to talks at the 2012 PDL Retreat.

To demonstrate the utility of request-flow comparison in various distributed systems, this dissertation describes its implementation in a tool called Spectroscope, and describes how Spectroscope was used to diagnose real, previously unsolved problems in the Ursa Minor distributed storage service and in select Google services. It also explores request-flow comparison's applicability to the Hadoop File System. Via 26-person user study, it identifies effective visualizations for presenting request-flow comparison's results, and further demonstrates that request-flow comparison helps developers quickly identify starting points for diagnosis. Finally, this dissertation distills end-to-end tracing design choices that will maximize a tracing infrastructure's utility for diagnosis tasks and other use cases.

DISSERTATION ABSTRACT: Algorithms for Large-Scale Astronomical Problems

Bin Fu

Carnegie Mellon University SCS Ph.D. Dissertation, May 2, 2013

Modern astronomical datasets are getting larger and larger, which already include billions of celestial objects and take up terabytes of disk space, and are expected to continue growing in the near future. Meanwhile, many existing solutions do not scale well to such large amount of data, which raises the following question: How can we use modern computer science techniques to help astronomers better analyze large datasets?

To answer this question, we apply various computer science techniques to provide fast and scalable solutions. We develop algorithms to better handle big data; we make use of database techniques to store and retrieve data; to distribute computation, we process large datasets using modern distributed computing frameworks, and analyze the characteristics of different frameworks (MPI, MapReduce, and GPU).

All the developed techniques are designed to work on datasets with billions astronomical objects. We have tested them extensively and report the improved running time in this thesis. We believe the interdisciplinary between computer science and astronomy has great potential, especially with more data involved in the future.

DISSERTATION ABSTRACT: Statistical Diagnosis of Chronic Problems in Production Systems

Soila Pertet Kavulya

Carnegie Mellon University ECE Ph.D. Dissertation, May 1, 2013

Large production systems are susceptible to chronic problems—performance degradations or exceptions that occur intermittently or affect a subset of end-users. Traditional approaches for diagnosis typically rely on a bottom-up approach to localize problems by correlating low-level alarms (such as resource utilization indicators or network packet loss) across components in a production system. However, these alarm-correlation approaches fall short when diagnosing chronics because they fail to provide the necessary application-level visibility to detect chronics effectively. Due to the scale and complexity of production systems, there can be multiple unresolved chronics at any given time, and these chronics are sometimes triggered by complex corner cases.

This dissertation presents a holistic framework for diagnosing chronics in production systems that relies on a suite of statistical tools to detect user-visible symptoms of problems, such

continued on page 13

continued from page 12

as slow requests, and drill-down on the root-cause of chronic problems by analyzing unmodified application-level and system-level logs. The use of unmodified logs makes our framework amenable for use in production systems where we may not have the luxury of modifying existing instrumentation. The framework comprises of the four components. First, an extensible log-analysis tool extracts end-to-end causal flows using the existing application-logs in the production system; these end-to-end flows capture the user's experience with the system. Second, anomaly-detection tools label each end-to-end flow as successful or failed. The anomaly-detection tools combine heuristics with a peer-comparison approach to identify odd-man-out behavior among peers. Third, a top-down statistical diagnostic tool combines multiple instrumentation sources to localize the root-cause of the problem by identifying attributes that are more correlated with failed flows than successful flows. Fourth, a visualization tool exploits peer-comparison to highlight anomalous nodes in a cluster.

The diagnostic framework has been used to localize real incidents at an academic cloud-computing cluster that runs the Hadoop parallel-processing framework, and a production Voice-over-IP system at a large telecommunications provider.

DISSERTATION ABSTRACT:
Scale and Concurrency of Massive File System Directories

Swapnil V. Patil

Carnegie Mellon University SCS Ph.D. Dissertation, April 30, 2013

File systems store data in files and organize these files in directories. Over decades, file systems have evolved to handle increasingly large files: they distribute files across a cluster of machines, they parallelize access to these files, they decouple data access from metadata access, and hence they provide scalable file access for high-performance applications. Sadly, most cluster-wide file systems lack any sophisticated support for large directories. In fact, most cluster file systems continue to use directories that were designed for humans, not for large-scale applications. The former use-case typically involves hundreds of files and infrequent concurrent mutations in each directory, while the latter use-case consists of tens of thousands of concurrent threads that simultaneously create large numbers of small files in a single directory at very high speeds. As a result, most cluster file systems exhibit very poor file create rate in a directory either due to limited scalability from using a single centralized directory server or due to reduced concurrency from using a system-wide synchronization mechanism.

Michelle Mazurek presents her research on "Usable Access Control for Personal Data" at the 2012 Retreat.

This dissertation proposes a directory architecture called GIGA+ that enables a directory in a cluster file system to store millions of files and sustain hundreds of thousands of concurrent file creations every second. GIGA+ makes two contributions: a concurrent indexing technique to scale-out a directory on many servers and an efficient layered design to scale-up performance. GIGA+ uses a hash-based, incremental partitioning algorithm that enables highly concurrent directory indexing through asynchrony and eventual consistency. This dissertation analyzes several trade-offs between data migration overhead, load balancing effectiveness, directory scan performance, and entropy of indexing state made by the GIGA+ design, and compares them with policies used in other systems. GIGA+ also demonstrates a modular implementation that separates directory distribution from directory representation. It layers a client-server middleware, which spreads work among many GIGA+ servers, on top of an backend storage system, which manages on-disk directory representation. This dissertation studies how system behavior is tightly dependent on both the indexing scheme and the on-disk implementations, and evaluates the performance for numerous backend stores including local and shared-disk systems. The GIGA+ prototype delivers highly scalable directory performance (that exceeded the most demanding Petascale-era requirements), provides the traditional UNIX file system interface (that can run applications without any modifications) and offers a new layered functionality on existing cluster file systems (that lack support for distributed directories).

DISSERTATION ABSTRACT:
Chaining for Flexible and High-Performance Key-Value Systems

Amar Phanishayee

Carnegie Mellon University SCS Ph.D. Dissertation, September 17, 2012

Distributed key-value (KV) systems are a critical part of the infrastructure at many large sites such as Amazon, Facebook, Google, and Twitter. Unfortunately, the ecosystem of these KV systems is a mess—no one existing system meets the needs of all applications. Systems designers worry about running multiple stores from different codebases, vendors, and so on, each optimized for certain application requirements and hardware configuration. We argue that having systems designers worry about running mul-

continued on page 14

continued from page 13

tiple stores from different codebases, vendors, and so on, each optimized for certain application requirements and hardware configuration, is unreasonable and unnecessary.

This dissertation proposes a key-value architecture using a generalization of chain-based replication which can be easily configured to support many points along the KV design continuum. First, we present a new replication protocol, Ouroboros, which extends chain-based replication to allow node additions to any part of the replica chain, minimize blocking during node additions and deletions, and guarantee provably strong data consistency. We use Ouroboros in the implementation of a distributed key-value storage system, FAWN-KV, designed with the goal of supporting the three key properties of fault tolerance, high performance, and generality. Second, we present a generalization of chain-based replication to effectively support a wide range of application requirements using four simple knobs: (a) replica type; (b) replication factor; (c) update mechanism between replicas; and (d) query node selection. We describe Flex-KV, that extends Ouroboros with this generalization. Flex-KV can support DRAM, Flash, and disk-based storage; can act as an unreliable cache or a durable store; and can offer strong or weak data consistency. 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.

Elie Krevat presents his research on "Adaptive Resource Allocation in Shared Service Environments" at the 2012 Retreat.

DISSERTATION ABSTRACT:
Understanding and Managing Propagation on Large Networks—Theory, Algorithms, and Models

B. Aditya Prakash

Carnegie Mellon University SCS Ph.D. Dissertation, September 18, 2012

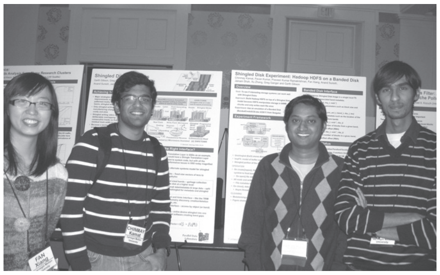
How do contagions spread in population networks? What happens if the networks are dynamic? Which hospitals should we give vaccines to, for maximum effect? How to detect sources of rumors on Twitter/Facebook?

These questions and many others such as which group should we market to, for maximizing product penetration, how quickly news travels in online media and how the relative frequencies of competing tasks evolve are all related to propagation/cascade-like phenomena on networks.

In this thesis, we present novel theory, algorithms and models for propagation processes on large static and dynamic networks, focusing on:

1. Theory: We tackle several fundamental questions like determining if there will be an epidemic, given the underlying networks and virus propagation models and predicting who-wins when viruses (or memes or products etc.) compete. We give a unifying answer for the threshold based on eigenvalues, and prove the surprising "winner-takes-all" result and other subtle phase-transitions for competition among viruses.

2. Algorithms: Based on our analysis, we give dramatically better algorithms for important tasks like effective immunization and reliably detecting culprits of epidemics. Thanks to our carefully designed algorithms, we achieve 6x fewer infections on real hospital patient-transfer graphs while also being significantly faster than other competitors (up to 30,000x).



From l to r, Fan Xiang, Chinmay Kamat, Pavan Kumar Alampalli and Praveen Kumar Ramakrishnan, all INI MS students, are ready to talk about their research.

3. Models: Finally using our insights, we study numerous datasets to develop powerful general models for information diffusion and competing species in a variety of situations. Our models unify earlier patterns and results, yet being succinct and enable challenging tasks like trend forecasting, spotting outliers and answering 'what-if' questions.

Our inter-disciplinary approach has led to many discoveries in this thesis, with broad applications spanning areas like public health, social media, product marketing and networking. We are arguably the first to present a systematic study of propagation and immunization of single as well as multiple viruses on arbitrary, real and time-varying networks as the vast majority of the literature focuses on structured topologies, cliques, and related unrealistic models.

DISSERTATION ABSTRACT:
Mining and Modeling Real-world Networks: Patterns, Anomalies, and Tools

Leman Akoglu

Carnegie Mellon University SCS Ph.D. Dissertation, August 22, 2012

Large real-world graph (a.k.a. network, relational) data are omnipresent, in online media, businesses,

continued on page 15

continued from page 14

science, and the government. Analysis of these massive graphs is crucial, in order to extract descriptive and predictive knowledge with many commercial, medical, and environmental applications. In addition to its general structure, knowing what stands out, i.e. anomalous or novel, in the data is often at least, or even more important and interesting. In this thesis, we build novel algorithms and tools for mining and modeling large-scale graphs, with a focus on:

(1) Graph pattern mining: we discover surprising patterns that hold across diverse real-world graphs, such as the "fortification effect" (e.g. the more donors a candidate has, the super-linearly more money s/he will raise), dynamics of connected components over time, and power-laws in human communications,

(2) Graph modeling: we build generative mathematical models, such as the RTG model based on "random typing" that successfully mimics a long list of properties that real graphs exhibit,

(3) Graph anomaly detection: we develop a suite of algorithms to spot abnormalities in various conditions; for (a) plain weighted graphs, (b) binary and categorical attributed graphs, (c) time-evolving graphs, and (d) sense-making and visualization of anomalies.

THESIS PROPOSAL:
More Consensus with Egalitarian Parliaments

Iulian Moraru, SCS

February 4, 2013

This thesis describes the design and implementation of state machine replication (SMR) protocols that achieve near-perfect load balancing and availability, near-optimal request processing latency (especially in the wide area), and performance robustness when confronted with failures and slow replicas. In the process, we attempt to root practical SMR implementation aspects such as time leases, failure detection and reconfiguration, that have traditionally been considered out of the scope of SMR protocols, in stable theoretical grounding.

At the center of our work is a new variant of the Paxos protocol that we call Egalitarian Paxos. In Egalitarian Paxos all replicas perform the same functions simultaneously to ensure better load balancing and availability, lower commit latency and higher performance robustness when compared to previous Paxos variants. We show—both theoretically and empirically—that Egalitarian Paxos has the aforementioned benefits and then extend its design with techniques that give it higher performance and availability, and lower resource utilization in a variety of practical scenarios, as well as the ability to tolerate Byzantine faults.

THESIS PROPOSAL:
A Tag-Based, Logical Access-Control Framework for Personal Data Sharing

Michelle Lynn Mazurek, ECE

December 17, 2012

Computer users are storing and sharing ever-increasing numbers of documents, photos, and other content, both on a multitude of personal devices and within online networks. In this environment, proper access control is critical to help users share varied content with different groups of people while avoiding trouble at work, embarrassment, identity theft, and other problems related to unintended disclosure. Correctly managing access control, however, has historically proven difficult, time-consuming, and error-prone, even for experts; to make matters worse, it remains a secondary task most non-expert users are unwilling to spend significant time on.

Garth Gibson accepts an award from Gary Grider (LANL), given in thanks for helping the DOE with the PLFS technology that is now part of the Exascale fast forward technology roadmap.

To solve this problem, online and distributed content-sharing services should provide an access-control system that provides verifiable security, makes policy configuration and management simple and understandable for users, reduces the risk of user error, and minimizes the required user effort. In the proposed thesis, we develop a novel access-control mechanism, which incorporates logic-based access-control techniques as well as tag-based policy specification, designed to meet these goals. We base our design on three user studies that provide insight into people's access-control needs and preferences. We implement this mechanism within an experimental distributed file system called Penumbra, in the process addressing key design challenges that arise from the combination of logic-based access control with tag-based policy specification. We evaluate Penumbra using a set of grounded case studies drawn primarily from user study results, demonstrating that its performance is reasonable for common scenarios. We also present a tradeoff analysis quantifying Penumbra's sensitivity to important differences in use cases.

THESIS PROPOSAL:
Building Memory-efficient Key-Value Clusters with Provable Bounds Using Better Hashing and Selected Caching

Bin Fan, SCS

December 10, 2012

Distributed systems have grown rapidly in scale, making cost-effective storage and access both more important

continued on page 16

continued from page 15

and more challenging. In this thesis, we propose techniques grounded in recent theory (e.g., cuckoo hashing and randomized load balancing) and informed by the underlying hardware and expected workloads, to design a cluster with substantially reduced per-node resource consumption and also achieves a near-optimal capacity utilization across nodes.

As a case study, we aim to build a distributed key-value (KV) cluster that achieves high performance for a broad variety of (or even adversarial) workloads, while still imposing low memory overhead. To this end, we will tackle two research problems:

First, how to create a single-node key-value store that delivers high throughput with minimal memory overhead? This thesis describes two state-of-the-art single-node key-value stores: one is based on flash storage to provide cost-effective data access and optimize for servers with limited CPU and memory capacity; the other uses memory as main storage for high throughput and low-latency retrieval for data that is transient and fits in memory. Second, how to build a many-node key-value cluster that avoids hot-spots at one or a few nodes under uneven or dynamic workloads? This thesis analytically proves and empirically demonstrates that a simple caching scheme complemented by randomized load balancing can effectively harness the imbalanced load across nodes, regardless of the distribution of query popularity.

John Dobyns (l) and Tom Ambrose (r), both of Emulex, enjoy a poster session at the 2012 PDL Retreat.

Our solutions greatly benefit from a set of novel techniques that are theory-grounded and optimized by architectural-aware and workload-inspired observations, including:

1. A memory-efficient, high-performance hash table based on cuckoo hashing.

2. A compact data structure for probabilistic set-membership testing, similar to Bloom filters but asymptomatically smaller if the false positive rate <2%, and with support to delete.

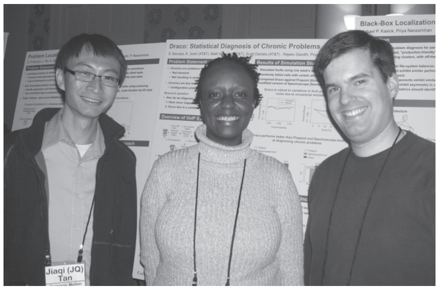
3. Provable load balancing: we prove that, in a n-node key-value cluster using hashing for key space partitioning (e.g., consistent hashing), deploying a small but fast caching service for the O(nlogn) most popular entries can effectively prevent individual nodes becoming hot-spots and thus ensure provable load balancing across the cluster.

THESIS PROPOSAL: Trading Latency for Freshness in Storage Systems

Jim Cipar, SCS

October 2012

Many storage systems have to provide extremely high throughput updates and low latency read queries. In practice, system designs that provide those capabilities often face a trade-off between query latency, efficiency, and result freshness. In my dissertation, I will argue that systems should be designed to allow for a per-query configuration of this trade-off. I will use two case studies to demonstrate the value of doing so. The first is LazyBase, a database designed for high-throughput ingest of observational data. The second is LazyTables, a shared data structure designed to support parallel machine learning applications. In both cases, the term "Lazy" refers to the systems' procrastination: waiting to apply updates until they can be executed as efficiently as possible. This design decision creates the potential for staleness in the data, hence the need for studying the trade-off between freshness and performance. Additionally, I will describe a number of other applications where this trade-off is potentially useful in system design.



From l to r, Jiaqi Tan, Soila Pertet Kavulya, and Mike Kasick are ready to present their research via a poster session at the 2012 PDL Retreat. All three are advised by Priya Narasimhan.

M.S. THESIS: Landslide: Systematic Dynamic Race Detection in Kernel-space

Ben Blum, SCS

May 10, 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. 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.

continued from page 1

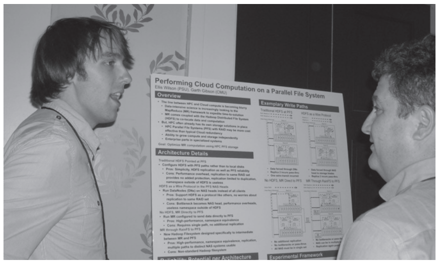
the network almost always turns out to be the limiting factor. As the machine count increases, the fraction of time spent on communication and synchronization grows compared to actual computation — in other words, computational threads spend more time waiting on communication (of necessary global parameters), and less time doing actual work.

If an algorithm is bottlenecked by the network, what can we possibly do to alleviate it? A key insight is that many important ML algorithms are *iterative-convergent*, repeating a set of operations (such as sampling or gradient steps) until an objective function stops changing. Such algorithms are often tolerant to small errors in their intermediate state (even the global parameters), as has been demonstrated both empirically and theoretically by the ML community. This tolerance presents an interesting opportunity: what if we could reduce network communication at the expense of increased error in the algorithm's state? Just as importantly, how do we ensure this error remains small enough for algorithm convergence?

The LazyTables System for Distributed Parameter Storage

Our answer is a global parameter server we call "LazyTables", a distributed system that allows shared parameters/variables to be read in a stale manner by many threads across multiple machines, and whose table-based interface requires minimal effort to accommodate existing ML algorithms.

LazyTables consists of a client library and a distributed server program. The server program manages a collection of tables, which hold intermediate state or global parameters used by ML algorithms. Server processes are distributed over multiple machines, and each table is sharded (divided) across all server processes. The client library is used to perform operations on the distributed tables, and it can be invoked from multiple processes on different machines and from multiple threads within the same process.



Ellis Wilson describes his research on "Performing Cloud Computation on a Parallel File System" to Jeff Heller of NetApp.

A typical LazyTables setup consists of *M* machines, where each machine runs both a LazyTables server process and a client process for executing the ML algorithm. The client process runs multiple computational threads (usually one per core of the multicore machine), which use the LazyTables client library to query the servers for table operations. Figure 1 illustrates such a setup.

At its core, LazyTables functions much like a distributed key-value store, but where variables are organized into table rows with multiple entries. This table abstraction allows the programmer to group related variables into rows, which are then managed by the LazyTables system. In addition to creating tables, clients may perform the following table operations:

- `Increment(table,row,element,value)`: Increases a table-row-element by value, a (possibly negative) floating point or integer number.
- `Put(table,row,element,value)`: Overwrite a table-row-element with value.
- `Read row(table,row,staleness)`: Retrieve a table-row given a staleness threshold. Staleness is the key concept of LazyTables, and will be explained shortly.
- `Refresh row(table,row,staleness)`: Prefetch a table-row in the background, resulting in faster table performance provided the programmer knows which rows will be needed in advance.
- `Clock()`: Increases the "clock time" of the client thread by one. A unit of clock time represents a programmer-defined quantity of computational work, and is a key part of the staleness concept.

Increments and puts have the "read-my-writes" property, meaning that increments/puts by a client thread will be visible in its next row read or increment (except when another thread overwrites the value with a put). This eliminates the need to manually track local state changes in a separate data structure.

Staleness

Staleness is the key feature of LazyTables, and basically permits servers to provide clients with out-of-date row values, up to a maximum permitted row age that clients can specify. When a client requests stale data, LazyTables can serve a row from a thread-local or process-local cache instead of the actual machine hosting the row, thus freeing precious network bandwidth for more important communication. Distributed algorithms spend much time waiting on communication with other machines; hence, requesting stale data from local caches will not only dramatically reduce the time to get the data, but also speed up other network communications by reducing network load.

LazyTables defines staleness in terms of client thread "clocks", where a unit of clock time represents a programmer-specified amount of computational work. For example, a programmer might decide to increment a client thread's clock every time it processes *C* data points. In that case, one unit of clock time represents all LazyTable increments/puts performed while processing those *C* points. The distributed servers keep track of all client clocks *c*, and when a client wants to perform a

continued on page 18

LAZYTABLES

continued from page 17

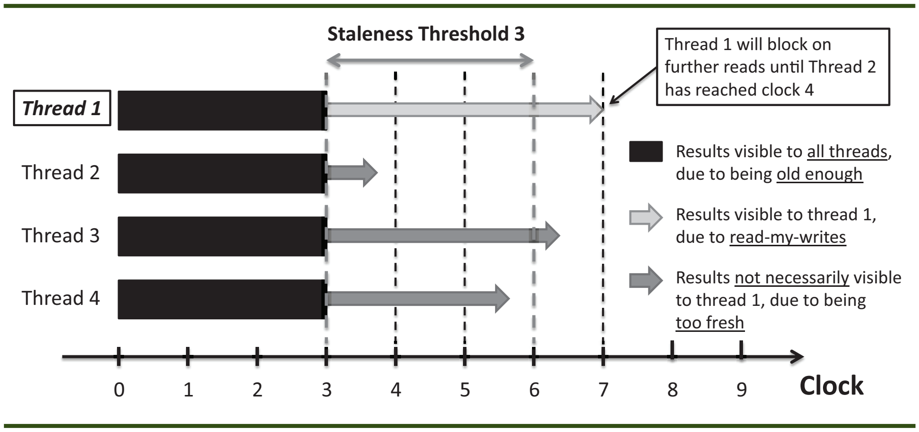

Staleness Threshold 3
Thread 1 will block on further reads until Thread 2 has reached clock 4
Thread 1
Thread 2
Thread 3
Thread 4
Results visible to all threads, due to being old enough
Results visible to thread 1, due to read-my-writes
Results not necessarily visible to thread 1, due to being too fresh
0
1
2
3
4
5
6
7
8
9
Clock

Figure 2. Bounded staleness in LazyTables: An example with 4 threads and staleness 3.

row read, the server tells the client that the row's clock r is the minimum of all other client clocks — in other words, the clock of the slowest client excluding itself. Figure 2 provides a graphical illustration of staleness.

Staleness is defined as "client clock c minus row clock r", i.e. how much older the row is compared to the client. When a client with clock c requests a row with staleness threshold s, it first checks its thread-level cache to see if the row is cached with clock $r_{thread} \ge c - s$, i.e. not older than the client by s. If so, the client reads the cached row and does not produce network traffic. Otherwise, the client checks the process-level cache for the same row, and if it is present with clock $r_{process} \ge c - s$, it fetches that row and avoids network communication. Finally, should the row be absent from both caches or is otherwise too stale, the client queries the server. If the server's row clock r_{server} is $\ge c - s$, then the client fetches the row, incurring network traffic. But if the row clock r_{server} is $< c - s$, this means there is another client thread whose clock has not reached $c - s$ yet. In this case, the requesting client must wait until all other threads have (at least) reached $c - s$.

The minimum possible staleness is $s = 0$ (which is roughly equivalent to the popular bulk synchronous parallel model), and there is no maximum limit. As a consequence of staleness, LazyTables has the following important properties:

1. A thread with clock c reading rows with staleness s must always wait for the slowest thread to reach clock $c - s$. This *bounded staleness* keeps all computational threads loosely synchronized in terms of work done.

2. When the slowest thread (in terms of clock) reads rows from the server, those rows' clocks will be ≥ its own clock. Hence, it can cache them longer than any other thread, before having to make another server read. Consequently, the slowest thread spends the least amount of time communicating over the network, *allowing it to catch up.*

3. Conversely, the fastest thread always reads rows with clock ≤ its own. Thus, it caches rows for a shorter duration than any other thread. As a result, it spends the most time on network communication, *allowing other threads to catch up.*

In contrast to systems that only permit clients to read the most recent version of parameters stored on the server, LazyTables client threads instead read from their thread-local or process-local caches to varying degrees. For example, the fastest thread rarely reads from its cache (because its fast-moving clock quickly makes its cached data too stale), while the slowest thread reads almost everything from its cache.

In summary, bounded staleness allows LazyTables to reduce and re-balance network traffic, with the following benefits:

- Faster threads (that have finished more work clocks), which have a greater need for fresh parameter values, receive more network bandwidth.
- Slower threads (that have finished fewer work clocks), also known as *stragglers*, spend less time on network communication, which allows them to catch up to faster threads while simultaneously reducing network traffic.

Using LazyTables, algorithms can achieve a high computation rate (due to self-balancing and network traffic reduction) without sacrificing too much convergence rate (because staleness is bounded). Our early experiments with multiple real machine learning algorithms and data sets show that LazyTables enables high iterations-per-time while maintaining reasonable convergence-per-iteration, resulting in higher convergence-per-time. We continue to explore this exciting new approach to efficiently supporting advanced Big Data computing.

References

[1] James Cipar, Qirong Ho, Jin Kyu Kim, Seunghak Lee, Gregory R. Ganger, Garth Gibson, Kimberly Keeton, Eric Xing. Solving the Straggler Problem with Bounded Staleness. 14th USENIX HotOS Workshop, Santa Ana Pueblo, NM, May 13-15, 2013.

The Big Learning Group is a collection of systems and machine learning researchers from PDL and elsewhere at Carnegie Mellon, including Jim Cipar, Henggang Cui, Wei Dai, Qirong Ho, Jin Kyu Kim, Seunghak Lee, Jinliang Wei, David Andersen, Greg Ganger, Phil Gibbons (Intel), Garth Gibson, Alex Smola and Eric Xing.

Bin Fu recording feedback from PDL Consortium Members at the 2012 Retreat.

continued from page 7

Practical Batch-Updatable External Hashing with Sorting

Lim, Andersen & Kaminsky

Meeting on Algorithm Engineering and Experiments (ALENEX), January 2013.

This paper presents a practical external hashing scheme that supports fast lookup (7 microseconds) for large datasets (millions to billions of items) with a small memory footprint (2.5 bits/item) and fast index construction (151 K items/s for 1-KiB key-value pairs). Our scheme combines three key techniques: (1) a new index data structure (Entropy-Coded Tries); (2) the use of sorting as the main data manipulation method; and (3) support for incremental index construction for dynamic datasets. We evaluate our scheme by building an external dictionary on flash-based drives and demonstrate our scheme's high performance, compactness, and practicality.

Giga+TableFS on PanFS: Scaling Metadata Performance on Cluster File Systems

Kulkarni, Ren, Patil & Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-13-101, January 2013.

Modern File Systems provide scalable performance for large file data management. However, in case of metadata management the usual approach is to have single or few points of metadata service (MDS). In the current world, file systems are challenged by unique needs such as managing exponentially growing files, using filesystem as a key-value store, checkpointing that are highly metadata intensive and are usually bottlenecked by the centralized MDS schemes.

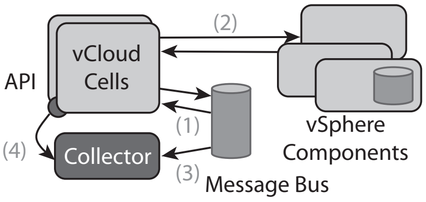
To overcome this metadata bottleneck, we evaluate a scalable MDS layer for the existing cluster file systems using Giga+ – a high performance distributed index without synchronization and serialization and TableFS – a file system with an embedded No-SQL database using modern key-value pair levelDB. We take layered approach to scale the metadata performance which does not need any hardware infrastructure upgrade in the existing storage clusters. In addition to providing scalable and increased metadata performance by several folds, avoiding metadata hotspots, packing small files, our MDS layer adds no-or-low performance overhead on the data throughput and resource utilizations of the underlying cluster.

vQuery: A Platform for Connecting Configuration and Performance

Shafer, Gylfason & Ganger

VMware Labs Technical Report, Palo Alto, CA. December 2012.

Discovering the causes of performance problems in virtualized systems is often more difficult than without virtualization, because they can be caused by changes in infrastructure configuration rather than the user's application. vQuery is a system that collects, archives, and exposes configuration changes alongside fine-grained performance data, so the two can be correlated. It gathers configuration change data without modifying the systems it collects from and copes with platform-specific details within a general, graph-based model of Infrastructure-as-a-Service (IaaS) infrastructures. Configuration data collected from two VMware® vSphereTM environments reveals that configuration changes are frequent and involved, opening interesting new directions for configuration-aware performance diagnosis techniques.



Configuration collection strategy for vCloud and OpenStack. The configuration collector listens to messages on the vCloud message bus and polls an appropriate API upon intercepting a task completion message. vCloud sends a message to start an action (e.g, start a VM (1)), which results in a message sent to an AMQP message bus (1) and actions in vSphere (2). When the task is finished, a completion message is posted to the message bus. The configuration collector listens to the same AMQP message bus (3), filters to listen to only task completion messages, and queries an appropriate API to find details about configuration change after a task completes (4).

Helping Users Create Better Passwords

Ur, Kelley, Komanduri, Lee, Maass, Mazurek, Passaro, Shay, Vidas, Bauer, Christin, L. Cranor, Egelman & López

USENIX ;login: 37(6), December 2012.

Over the past several years, we have researched how passwords are created, how they resist cracking, and how usable they are. In this article, we focus on recent work in which we tested various techniques that may encourage better password choices. What we found may surprise you.

Despite a litany of proposed password replacements, text-based passwords are not going to disappear anytime soon [4]. Passwords have a number of advantages over other authentication mechanisms. They are simple to implement, relatively straightforward to revoke or change, easy for users to understand, and allow for quick authentication; however, passwords also have a number of drawbacks. Foremost among these drawbacks is that it is difficult for users to create and remember passwords that are hard for an attacker to guess. Our research group at Carnegie Mellon University has been investigating strategies to guide users to create passwords that are both secure and memorable.

continued on page 20

continued from page 19

Theia: Visual Signatures for Problem Diagnosis in Large Hadoop Clusters

Garduno, Kavulya, Tan, Gandhi & Narasimhan

26th Large Installation System Administration Conference (LISA'12), Dec. 9-14, San Diego, CA. Best Student Paper. Also published in USENIX ;login, 38(2), April 2013.

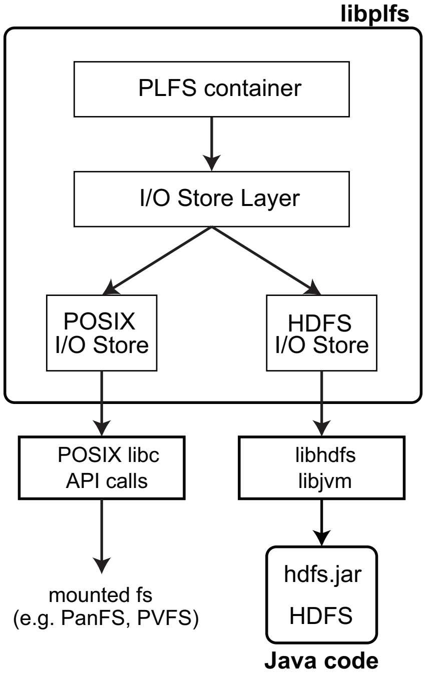
Diagnosing performance problems in large distributed systems can be daunting as the copious volume of monitoring information available can obscure the root-cause of the problem. Automated diagnosis tools help narrow down the possible root-causes—however, these tools are not perfect thereby motivating the need for visualization tools that allow users to explore their data and gain insight on the root-cause. In this paper we describe Theia, a visualization tool that analyzes application-level logs in a Hadoop cluster, and generates visual signatures of each job's performance. These visual signatures provide compact representations of task durations, task status, and data consumption by jobs. We demonstrate the utility of Theia on real incidents experienced by users on a production Hadoop cluster.

HPC Computation on Hadoop Storage with PLFS

C. Cranor, Polte & Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-115. November 2012.

In this report we describe how we adapted the Parallel Log Structured Filesystem (PLFS) to enable HPC applications to be able read and write data from the HDFS cloud storage subsystem. Our enhanced version of PLFS provides HPC applications with the ability to concurrently write from multiple compute nodes into a single file stored in HDFS, thus allowing HPC applications to checkpoint. Our results show that HDFS combined with our PLFS HDFS I/O Store module is able to handle a concurrent write checkpoint workload generated by a benchmark with good performance.



PLFS I/O Store architecture.

Failure Diagnosis of Complex Systems

Kavulya, Joshi (AT&T), Di Giandomenico (ISTI-CNR, Pisa, Italy) & Narasimhan

Chapter in "*Resilience Assessment and Evaluation.*" Editors. Katinka Wolter, Alberto Avritzer, Marco Vieira, Aad van Moorsel. Springer Verlag, December 2012.

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.

Runtime Estimation and Resource Allocation for Concurrency Testing

Simsa, Bryant & Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-113. December 2012.

In the past 15 years, stateless exploration, a collection of techniques for automated and systematic testing of concurrent programs, has experienced wide-spread adoption. As stateless exploration moves into practice, becoming part of testing infrastructures of large-scale system developers, new practical challenges are being identified.

In this paper we address the problem of efficient allocation of resources to stateless exploration runs. To this end, this paper presents techniques for estimating the total runtime of stateless exploration runs and policies for allocating resources among tests based on these runtime estimates.

Evaluating our techniques on a collection of traces from a real-world deployment at Google, we demonstrate the techniques' success at providing accurate runtime estimations, achieving estimation accuracy above 60% after as little as 1% of the state space has been explored. We further show that these

continued on page 21

continued from page 20

estimates can be used to implement intelligent resource allocation policies that meet testing objectives more than twice as efficiently as the round-robin policy.

TABLEFS: Embedding a NoSQL Database inside the Local File System

Ren & Gibson

1st Storage System, Hard Disk and Solid State Technologies Summit, IEEE Asia-Pacific Magnetic Recording Conference (APMRC), November 2012, Singapore.

Conventional file systems are optimized for large file transfers instead of workloads that are dominated by metadata and small file accesses. This paper examines using techniques adopted from NoSQL databases to manage file system metadata and small files, which feature high rates of change and efficient out-of-core data representation. A FUSE file system prototype was built by storing file system metadata and small files into a modern key-value store: LevelDB. We demonstrate that such techniques can improve the performance of modern local file systems in Linux for workloads dominated by metadata and tiny files.

A Case for Scaling HPC Metadata Performance through De-specialization

Patil, Ren & Gibson

7th Petascale Data Storage Workshop held in conjunction with Supercomputing '12, November 12, 2012. Salt Lake City, UT. Supersedes Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-111, November 2012.

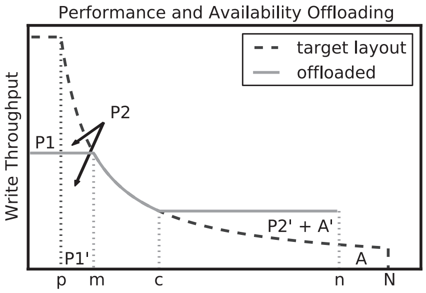
Modern cluster file systems provide highly scalable I/O bandwidth along the data path by enabling highly parallel access to file data. Unfortunately metadata scaling is lagging behind data scaling. We propose a file system design that inherits the scalable data bandwidth of existing cluster file systems and adds support for distributed and high-performance metadata operations. Our key idea is to integrate a distributed indexing mechanism with general-purpose optimized on-disk metadata store. Early prototype evaluation shows that our approach outperforms popular Linux local file systems and scales well with large numbers of file creations.

JackRabbit: Improved Agility in Elastic Distributed Storage

Cipar, Xu, Krevat, Tumanov, Gupta, Kozuch & Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-112, October 2012.

Elastic storage systems can be expanded or contracted to meet current demand, allowing servers to be turned off or used for other tasks. However, the usefulness of an elastic distributed storage system is limited by its agility: how quickly it can increase or decrease its number of servers. This paper describes an elastic storage system, called JackRabbit, that can quickly change its number of active servers. JackRabbit uses a combination of agility-aware offloading and reorganization techniques to minimize the work needed before deactivation or activation of servers. Analysis of real-world traces and experiments with the JackRabbit prototype confirm \sysname{}'s agility and show that it significantly reduces wasted server-time relative to state-of-the-art designs.



Write offloading for performance and availability. Writes for over-loaded primaries are load balanced across the offload set (servers 1 to *m*), and writes to non-primaries in the offload set (servers *p*+1 to *m*) are load-balanced over other servers. Writes for non-active servers are also load-balanced over other servers. Each such offloaded area is marked with a symbol and a symbol-prime, whose sizes must be equal (e.g., P1 is offloaded to P1').

HPC Computation on Hadoop Storage with PLFS

Cranor, Polte & Gibson

Carnegie Mellon University Parallel Data Laboratory Technical Report CMU-PDL-12-115, October 2012.

In this report we describe how we adapted the Parallel Log Structured Filesystem (PLFS) to enable HPC applications to be able read and write data from the HDFS cloud storage subsystem. Our enhanced version of PLFS provides HPC applications with the ability to concurrently write from multiple compute nodes into a single file stored in HDFS, thus allowing HPC applications to checkpoint. Our results show that HDFS combined with our PLFS HDFS I/O Store module is able to handle a concurrent write checkpoint workload generated by a benchmark with good performance.

Runtime Estimation of Stateless Exploration

Simsa, Bryant & Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-113, October 2012.

In the past 15 years, stateless exploration, a collection of techniques for automated and systematic testing of concurrent programs, has experienced a wide-spread adoption. As stateless exploration moves into practice, becoming part of testing infrastructure of large-scale system developers, new pragmatic challenges are being

continued on page 22

continued from page 21

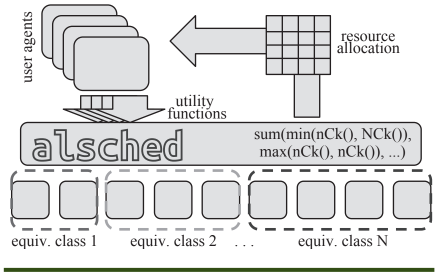
identified. In this report we address the problem of accurate runtime estimation of stateless exploration by designing techniques that address the non-linear nature through which modern stateless exploration techniques enumerate the state space of possible test executions and evaluate our techniques on a collection of exploration traces from a real-world deployment at Google.

alsched: Algebraic Scheduling of Mixed Workloads in Heterogeneous Clouds

Tumanov, Cipar, Kozuch & Ganger

3rd ACM Symposium on Cloud Computing. Oct. 14-17, 2012 - San Jose, CA.

As cloud resources and applications grow more heterogeneous, allocating the right resources to different tenants' activities increasingly depends upon understanding tradeoffs regarding their individual behaviors. One may require a specific amount of RAM, another may benefit from a GPU, and a third may benefit from executing on the same rack as a fourth. This paper promotes the need for and an approach for accommodating diverse tenant needs, based on having resource requests indicate any soft (i.e., when certain resource types would be better, but are not mandatory) and hard constraints in the form of composable utility functions. A scheduler that accepts such requests can then maximize overall utility, perhaps weighted by priorities, taking into account application specifics. Experiments with a prototype scheduler, called alsched, demonstrate that support for soft constraints is important for efficiency in multi-purpose clouds and that composable utility functions can provide it.



alsched System Model

Landslide: Systematic Exploration for Kernel-Space Race Detection

Blum & Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-114, October 2012.

Systematic exploration is an approach to finding race conditions by deterministically executing many thread interleavings and identifying which ones expose bugs. Current techniques are suitable for testing user-space programs, but are inadequate for testing operating system kernels. Testing kernel-level code necessitates understanding the kernel's design in order to effectively control nondeterminism and achieve reasonable state-space reduction. We present Landslide, a systematic exploration tool for finding races in kernels. Landslide makes use of user-provided configuration to enable efficient exploration and testing of meaningful interleavings. The user instruments the kernel to inform Landslide of important concurrency events, and configures Landslide's search to de-emphasize irrelevant kernel components. This combines the user's design knowledge with Landslide's ability to explore large state spaces. Our experience with Landslide shows that a tool built for this usage pattern is capable of identifying otherwise-overlooked kernel-space races.

A Case for Scaling HPC Metadata Performance through De-specialization

Patil, Ren & Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-111, October 2012.

Modern cluster file systems provide highly scalable I/O bandwidth along the data path by enabling highly parallel access to file data. Unfortunately metadata scaling is lagging behind data scaling. We propose a file system design that inherits the scalable data bandwidth of existing cluster file systems and adds support for distributed and high-performance metadata operations. Our key idea is to integrate a distributed indexing mechanism with general-purpose optimized on-disk metadata store. To demonstrate the feasibility of our approach, we implemented a prototype middleware layer using the FUSE file system and evaluated it on 64-node cluster. Preliminary results show promising scalability and performance: the single- node local metadata store was 10X faster than modern local file systems and the distributed middleware metadata service scaled well with a peak performance of 190,000 file creates per second on a 64-server configuration.

HAT: Heterogeneous Adaptive Throttling for On-Chip Networks

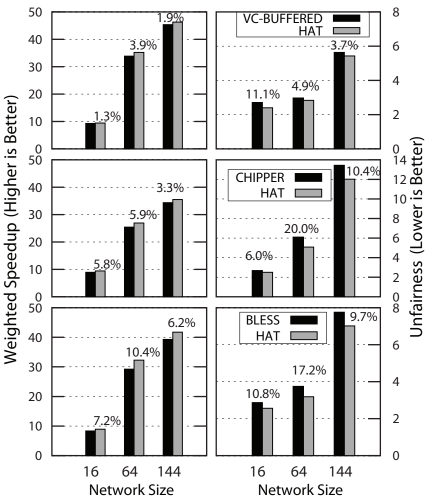
Chang, Ausavarungnirun, Fallin & Mutlu

SBAC-PAD 2012, New York, NY, October 24-26, 2012.

The network-on-chip (NoC) is a primary shared resource in a chip multiprocessor (CMP) system. As core counts continue to increase and applications become increasingly data-intensive, the network load will also increase, leading to more congestion in the network. This network congestion can degrade system performance if the network load is not appropriately controlled. Prior works have proposed source-throttling congestion control, which limits the rate at which new network traffic (packets) enters the NoC in order to reduce congestion and improve performance. These prior congestion control mechanisms have shortcomings that significantly limit their performance: either 1) they are not application-aware, but

continued on page 23

continued from page 22



System performance and fairness of CMP systems with varied network sizes and VC-buffered, CHIPPER, and BLESS routers.

rather throttle all applications equally regardless of applications' sensitivity to latency, or 2) they are not network-load-aware, throttling according to application characteristics but sometimes under- or over-throttling the cores.

In this work, we propose Heterogeneous Adaptive Throttling, or HAT, a new source-throttling congestion control mechanism based on two key principles: application-aware throttling and network-load-aware throttling rate adjustment. First, we observe that only network-bandwidth-intensive applications (those which use the network most heavily) should be throttled, allowing the other latency-sensitive applications to make faster progress without as much interference. Second, we observe that the throttling rate which yields the best performance varies between workloads; a single, static, throttling rate under-throttles some workloads while over-throttling others. Hence, the throttling mechanism should observe network load dynamically and adjust its throttling rate accordingly. While some past works have also used a closed-loop control approach, none have been application-aware. HAT is the first mechanism to combine application-awareness and network-load-aware throttling rate adjustment to address congestion in a NoC.

We evaluate HAT using a wide variety of multiprogrammed workloads on several NoC-based CMP systems with 16-, 64-, and 144-cores and compare its performance to two state-of-the-art congestion control mechanisms. Our evaluations show that HAT consistently provides higher system performance and fairness than prior congestion control mechanisms.

TABLEFS: Embedding a NoSQL Database Inside the Local File System

Ren & Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-110, September 2012.

File systems that manage magnetic disks have long recognized the importance of sequential allocation and large transfer sizes for file data. Fast random access has dominated metadata lookup data structures with increasingly use of B-trees on-disk. For updates, on-disk data structures are increasingly non-overwrite, copy-on-write, log-like and deferred. Yet our experiments with workloads dominated by metadata and small file access indicate that even sophisticated local disk file systems like Ext4, XFS and BTRFS leaves a lot of opportunity for performance improvement in workloads dominated by metadata and small files.

In this paper we present a simple stacked file system, TableFS, which uses another local file system as an object store and organizes all metadata into a single sparse table backed on-disk using a Log-Structured Merge (LSM) tree, LevelDB in our experiments. By stacking, TableFS asks only for efficient large file allocation and access from the local file system. By using an LSM tree, TableFS ensures metadata is written to disk in large, non-overwrite, sorted and indexed logs, and inherits a compaction algorithm. Even an inefficient FUSE based user level implementation of TableFS can perform comparably to Ext4, XFS and BTRFS on simple data-intensive benchmarks, and can outperform them by 50% to as much as 1000% for a metadata-intensive query/update workload on data-free files. Such promising performance results from TableFS suggest that local disk file systems can be significantly improved by much more aggressive aggregation and batching of metadata updates.

Row Buffer Locality Aware Caching Policies for Hybrid Memories

Yoon, Meza, Ausavarungnirun, Harding & Mutlu

Proceedings of the 30th IEEE International Conference on Computer Design (ICCD 2012), Montreal, Quebec, Canada, September 2012. Best paper award in Computer Systems and Applications track.

Phase change memory (PCM) is a promising technology that can offer higher capacity than DRAM. Unfortunately, PCM's access latency and energy are higher than DRAM's and its endurance is lower. Many DRAM-PCM hybrid memory systems use DRAM as a cache to PCM, to achieve the low access latency and energy, and high endurance of DRAM, while taking advantage of PCM's large capacity. A key question is what data to cache in DRAM to best exploit the advantages of each technology while avoiding its disadvantages as much as possible.

We propose a new caching policy that improves hybrid memory performance and energy efficiency. Our observation is that both DRAM and PCM banks employ row buffers that act as a cache for the most recently accessed memory row. Accesses that are row buffer hits incur similar latencies (and energy consumption) in DRAM and PCM, whereas accesses that are row buffer misses incur longer latencies

continued on page 24

continued from page 23

(and higher energy consumption) in PCM. To exploit this, we devise a policy that avoids accessing in PCM data that frequently causes row buffer misses because such accesses are costly in terms of both latency and energy. Our policy tracks the row buffer miss counts of recently used rows in PCM, and caches in DRAM the rows that are predicted to incur frequent row buffer misses. Our proposed caching policy also takes into account the high write latencies of PCM, in addition to row buffer locality.

Compared to a conventional DRAM-PCM hybrid memory system, our row buffer locality-aware caching policy improves system performance by 14% and energy efficiency by 10% on data-intensive server and cloud-type workloads. The proposed policy achieves 31% performance gain over an all-PCM memory system, and comes within 29% of the performance of an all- DRAM memory system (not taking PCM's capacity benefit into account) on evaluated workloads.

A Proof of Correctness for Egalitarian Paxos

Moraru, Andersen & Kaminsky

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-109. September 2012.

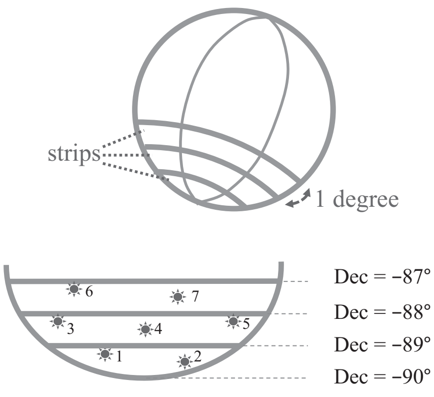
This paper presents a proof of correctness for Egalitarian Paxos (EPaxos), a new distributed consensus algorithm based on Paxos. EPaxos achieves three goals: (1) availability without interruption as long as a simple majority of replicas are reachable—its availability is not interrupted when replicas crash or fail to respond; (2) uniform load balancing across all replicas—no replicas experience higher load because they have special roles; and (3) optimal commit latency in the wide-area when tolerating one and two failures, under realistic conditions. Egalitarian Paxos is to our knowledge the first distributed consensus protocol to achieve all of these goals efficiently: requiring only a simple majority of replicas to be non-faulty, using a number of messages linear in the number of replicas to choose a command, and committing commands after just one communication round (one round trip) in the common case or after at most two rounds in any case.

Indexing and Fast Near-Matching of Billions of Astronomical Objects

Fu, Fink, Gibson & Carbonell

Proceedings of the Fourth Workshop on Interfaces and Architecture for Scientific Data Storage, 2012 (IASDS12). September 24, 2012, Beijing, China.

When astronomers analyze sky images, they need to identify the newly observed celestial objects in the catalog of known objects. We have developed a technique for indexing catalogs, which supports fast retrieval of closely matching catalog objects for every object in new images. It allows processing of a sky image in less than a second, and it scales to catalogs with billions of objects.



Indexing procedure. Top: The celestial sphere is divided into one-degree-wide strips. Bottom: We assume that there are seven objects in the catalog in this example, which are distributed among three strips. For each strip, the objects within the strip are sorted by their right ascension, and stored as a separate file.

Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis

Reiss, Tumanov, Ganger, Katz & Kozuch

3rd ACM Symposium on Cloud Computing. October 14th-17th, 2012 - San Jose, CA.

To better understand the challenges in developing effective cloud-based resource schedulers, we analyze the first publicly available trace data from a sizable multi-purpose cluster. The most notable workload characteristic is heterogeneity: in resource types (e.g., cores:RAM per machine) and their usage (e.g., duration and resources needed). Such heterogeneity reduces the effectiveness of traditional slot- and core-based scheduling. Furthermore, some tasks are constrained as to the kind of machine types they can use, increasing the complexity of resource assignment and complicating task migration. The workload is also highly dynamic, varying over time and most workload features, and is driven by many short jobs that demand quick scheduling decisions. While few simplifying assumptions apply, we find that many longer-running jobs have relatively stable resource utilizations, which can help adaptive resource schedulers.

How Does Your Password Measure Up? The Effect of Strength Meters on Password Creation

Ur, Kelley, Komanduri, Lee, Maass, Mazurek, Passaro, Shay, Vidas, Bauer, Christin & L. Cranor

In the 2012 USENIX Security Symposium, August 2012.

To help users create stronger text-based passwords, many web sites have deployed password meters that provide visual feedback on password strength. Although these meters are in wide use, their effects on the security and usability of passwords have

continued on page 25

continued from page 24

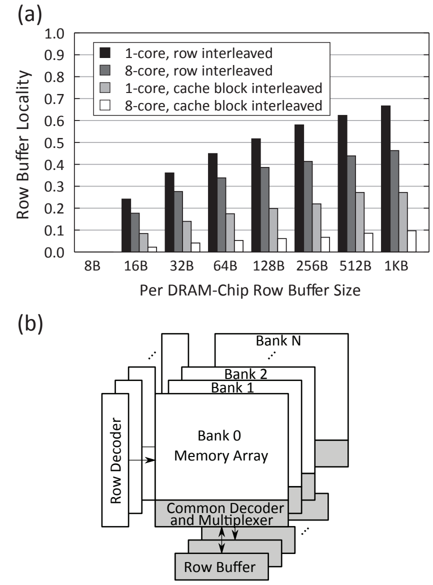
not been well studied. We present a 2,931-subject study of password creation in the presence of 14 password meters. We found that meters with a variety of visual appearances led users to create longer passwords. However, significant increases in resistance to a password-cracking algorithm were only achieved using meters that scored passwords stringently. These stringent meters also led participants to include more digits, symbols, and uppercase letters. Password meters also affected the act of password creation. Participants who saw stringent meters spent longer creating their password and were more likely to change their password while entering it, yet they were also more likely to find the password meter annoying. However, the most stringent meter and those without visual bars caused participants to place less importance on satisfying the meter. Participants who saw more lenient meters tried to fill the meter and were averse to choosing passwords a meter deemed "bad" or "poor." Our findings can serve as guidelines for administrators seeking to nudge users towards stronger passwords.

A Case for Small Row Buffers in Non-Volatile Main Memories

Meza, Li & Mutlu

Proceedings of the 30th IEEE International Conference on Computer Design (ICCD 2012), Poster Session, Montreal, Quebec, Canada, September 2012.

DRAM-based main memories have read operations that destroy the read data, and as a result, must buffer large amounts of data on each array access to keep chip costs low. Unfortunately, system-level trends such as increased memory contention in multi-core architectures and data mapping schemes that improve memory parallelism lead to only a small amount of the buffered data to be accessed. This makes buffering large amounts of data on every memory array access energy-inefficient; yet organizing DRAM chips to buffer small amounts of data is costly, as others have shown.



Row size affects row locality (a); our NVM architecture (b).

Emerging non-volatile memories (NVMs) such as PCM, STT-RAM, and RRAM, however, do not have destructive read operations, opening up opportunities for employing small row buffers without incurring additional area penalty and/or design complexity. In this work, we discuss and evaluate architectural changes to enable small row buffers at a low cost in NVMs. We find that on a multicore system, reducing the row buffer size can greatly reduce main memory dynamic energy compared to a DRAM baseline with large row sizes, without greatly affecting endurance, and for some NVM technologies, leads to improved performance.

Egalitarian Paxos

Moraru, Andersen & Kaminsky

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-108. July 2012.

This paper describes the design and implementation of Egalitarian Paxos (EPaxos), a new distributed consensus algorithm based on Paxos. EPaxos achieves two goals: (1) availability without interruption as long as a simple majority of replicas are reachable—its availability is not interrupted when replicas crash or fail to respond; and (2) uniform load balancing across all replicas—no replicas experience higher load because they have special roles. Egalitarian Paxos is to our knowledge the first distributed consensus protocol to achieve both of these goals efficiently: requiring only a simple majority of replicas to be non-faulty, using a number of messages linear in the number of replicas to choose a command, and committing commands after just one communication round (one round trip) in the common case or after at most two rounds in any case. We prove Egalitarian Paxos's properties theoretically and demonstrate its advantages empirically.

Staged Memory Scheduling: Achieving High Performance and Scalability in Heterogeneous Systems

Ausavarungnirun, Chang, Subramanian, Loh & Mutlu

The 39th International Symposium on Computer Architecture (ISCA), Portland, Oregon, June 9-13th, 2012.

When multiple processor (CPU) cores and a GPU integrated together on the same chip share the off-chip main memory, requests from the GPU can heavily interfere with requests from the CPU cores, leading to low system performance and starvation of CPU cores. Unfortunately, state-of-the-art application-aware memory scheduling algorithms are ineffective at solving this problem at low complexity due to the large amount of GPU traffic. A large and costly request buffer is needed to provide these algorithms with enough visibility across the global request stream, requiring relatively complex hardware implementations.

continued on page 26

continued from page 25

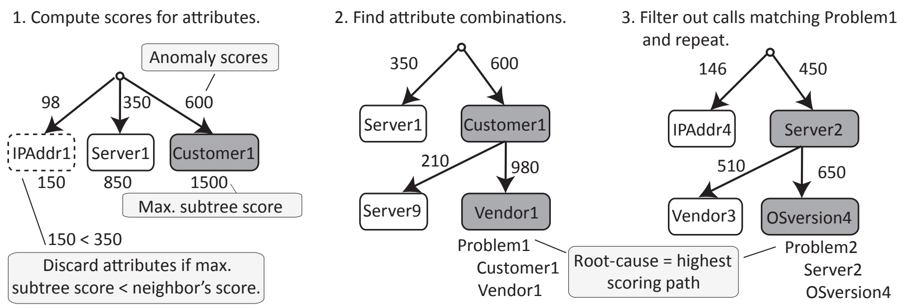
This paper proposes a fundamentally new approach that decouples the memory controller's three primary tasks into three significantly simpler structures that together improve system performance and fairness, especially in integrated CPU-GPU systems. Our three-stage memory controller first groups requests based on row-buffer locality. This grouping allows the second stage to focus only on inter-application request scheduling. These two stages enforce high-level policies regarding performance and fairness, and therefore the last stage consists of simple per-bank FIFO queues (no further command reordering within each bank) and straightforward logic that deals only with low-level DRAM commands and timing.

We evaluate the design trade-offs involved in our Staged Memory Scheduler (SMS) and compare it against three state-of-the-art memory controller designs. Our evaluations show that SMS improves CPU performance without degrading GPU frame rate beyond a generally acceptable level, while being significantly less complex to implement than previous application-aware schedulers. Furthermore, SMS can be configured by the system software to prioritize the CPU or the GPU at varying levels to address different performance needs.

Draco: Statistical Diagnosis of Chronic Problems in Large Distributed Systems

Kavulya, Daniels, Joshi, Hiltunen, Gandhi & Narasimhan.

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



Draco uses an iterative Bayesian approach to rank combinations of attributes most correlated with the problem.

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.

Light-weight Black-box Failure Detection for Distributed Systems

Tan, Kavulya, Gandhi & Narasimhan

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-107. July 2012.

Diagnosing failures in distributed systems is challenging, as modern datacenters run a variety of applications and systems. Current techniques for detecting failures often require training, have limited scalability, or are not intuitive to sysadmins. We present LFD, a lightweight and scalable technique for diagnosing performance problems in distributed systems using only correlations of operating system metrics collected transparently. The LFD fault detection algorithm is based on our hypothesis of server application behavior, and hence does not require training, and can perform failure detection with complexity linear in the number of nodes, with results that are intuitively interpretable by sysadmins. Further, with some training, LFD-DT uses decision-trees to diagnose the category of a problem that has previously been seen. We further show that LFD is versatile, and can diagnose faults in Hadoop MapReduce systems and on multi-tier web request systems, and show how LFD is intuitive to sysadmins.

Correct Horse Battery Staple: Exploring the Usability of System-assigned Passphrases

Shay, Kelley, Komanduri, Mazurek, Ur, Vidas, Bauer, Christin & L. Cranor

In SOUPS 2012: Symposium on Usable Privacy and Security, July 2012.

Users tend to create passwords that are easy to guess, while system-assigned passwords tend to be hard to remember. Passphrases, space-delimited sets of natural language words, have been

continued on page 27

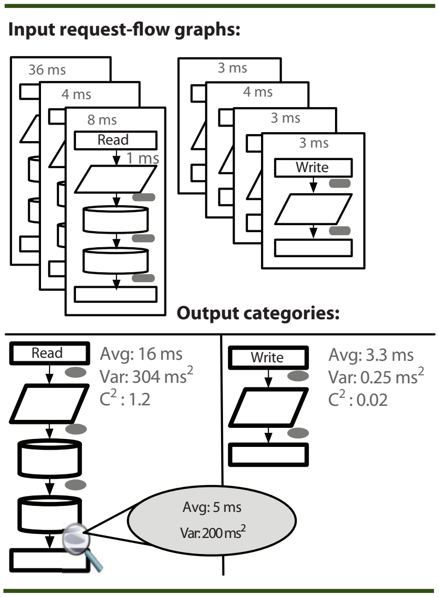
continued from page 26

suggested as both secure and usable for decades. In a 1,476-participant online study, we explored the usability of 3- and 4-word system-assigned passphrases in comparison to system-assigned passwords composed of 5 to 6 random characters, and 8-character system-assigned pronounceable passwords. Contrary to expectations, system- assigned passphrases performed similarly to system-assigned passwords of similar entropy across the usability metrics we examined. Passphrases and passwords were forgotten at similar rates, led to similar levels of user difficulty and annoyance, and were both written down by a majority of participants. However, passphrases took significantly longer for participants to enter, and appear to require error-correction to counteract entry mistakes. Passphrase usability did not seem to increase when we shrunk the dictionary from which words were chosen, reduced the number of words in a passphrase, or allowed users to change the order of words.

Automated Diagnosis without Predictability is a Recipe for Failure

Sambasivan & Ganger

Proceedings of the 4th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud '12), June 12-13, 2012, Automated management is critical to the success of cloud computing, given its scale and complexity. But, most systems do not satisfy one of the key properties required for automation: predictability, which in turn relies upon low variance. Most automation tools are not effective when variance is consistently high. Using automated performance diagnosis as a concrete example, this position paper argues that for automation to become a reality, system builders must treat variance as an important metric and make conscious decisions about where to reduce it. To help with this task, we describe a framework for reasoning about sources of variance in distributed systems and describe an example tool for helping identify them.



Example of how a VarianceFinder implementation might categorize requests to identify functionality with high variance. VarianceFinder assumes that requests that take the same path through a distributed system should incur similar costs. It groups request-flow graphs that exhibit the same structure into categories and calculates statistical metrics for them. Categories are ranked by the squared coefficient of variation (C^2) and high-variance edges along their critical path are automatically highlighted (as indicated by the magnifying glass).

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

Fu, Fink, Gibson & 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 18-19, 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.

Hadoop's Adolescence: A Comparative Workload Analysis from Three Research Clusters

Ren, Kwon, Balazinska, Howe

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-106. June 2012.

We analyze Hadoop workloads from three different research clusters from an application-level perspective, with two goals: (1) explore new issues in application patterns and user behavior and (2) understand key performance challenges related to IO and load balance. Our analysis suggests that Hadoop usage is still in its adolescence. We see underuse of Hadoop features, extensions, and tools as well as significant opportunities for optimization. We see significant diversity in application styles, including some

continued on page 28

continued from page 27

"interactive" workloads, motivating new tools in the ecosystem. We find that some conventional approaches to improving performance are not especially effective and suggest some alternatives. Overall, we find significant opportunity for simplifying the use and optimization of Hadoop, and make recommendations for future research.

SkyeFS: Distributed Directories using Giga+ and PVFS

Chivetta, Patil & Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-104, May 2012.

There is growing set of large-scale data-intensive applications that require file system directories to store millions to billions of files in each directory and to sustain hundreds of thousands of concurrent directory operations per second. Unfortunately, most cluster file systems are unable to provide this level of scale and parallelism. In this research, we show how the GIGA+ distributed directory algorithm, developed at CMU, can be applied to a real-world cluster file system. We designed and implemented a user-level file system, called SkyeFS, that efficiently layers GIGA+ on top of the PVFS cluster file system. Our experimental evaluation demonstrates how an optimized interposition layer can help PVFS achieve the desired scalability for massive file system directories.

Shingled Magnetic Recording for Big Data Applications

Suresh, Gibson & Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-12-105. May 2012.

Modern Hard Disk Drives (HDDs) are fast approaching the superparamagnetic limit forcing the storage industry to look for innovative ways to transition from traditional magnetic recording to Heat-Assisted Magnetic Recording or Bit-Patterned Magnetic Recording. Shingled Magnetic Recording (SMR) is a step in this direction as it delivers high storage capacity with minimal changes to current production infrastructure. However, since it sacrifices random-write capabilities of the device, SMR cannot be used as a drop-in replacement for traditional HDDs.

We identify two techniques to implement SMR. The first involves the insertion of a shim layer between the SMR device and the host, similar to the Flash Translation Layer found in Solid-State Drives (SSDs). The second technique, which we feel is the right direction for SMR, is to push enough intelligence up into the file system to effectively mask the sequential-write nature of the underlying SMR device. We present a custom-built SMR Device Emulator and ShingledFS, a FUSE-based SMR-aware file system that operates in tandem with the SMR Device Emulator. Our evaluation studies SMR for Big Data applications and we also examine the overheads introduced by the emulation. We show that Big Data workloads can be run effectively on SMR devices with an overhead as low as 2.2% after eliminating the overheads of emulation. Finally we present insights on garbage collection mechanisms and policies that will aid future SMR research.

Raja Sambasivan presents his research on "Visualizing Request-flow Comparison" at the 2012 PDL Retreat.

PDL Workshop and Retreat 2012.