PARALLEL DATA LABORATORY
CARNEGIE MELLON UNIVERSITY

PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • SPRING 2014
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

Table FS 1
Director's Letter 2
Year in Review 4
Recent Publications 5
PDL News & Awards 8
New PDL Faculty 10
Dissertations & Proposals 14

PDL CONSORTIUM MEMBERS

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

Enhancing Metadata Efficiency in the Local File System with TABLEFS

Kai Ren, Garth Gibson & Joan Digney

Even in the era of big data, most things in many file systems are small. File systems for magnetic disks have long suffered low performance when accessing huge collections of small files because of slow random disk seeks. Inevitably, scalable systems should expect the numbers of small files to achieve and exceed billions, but currently, effective scaling is not available for workloads that are dominated by metadata and tiny file access. Instead there has emerged a class of scalable small-data storage systems, commonly called key-value stores, which emphasize simple (NoSQL) interfaces and large in-memory caches.

Some of these key-value stores feature high rates of change and efficient out-of-memory Log-structured Merge (LSM) tree structures. An LSM tree can provide fast random updates, inserts and deletes without sacrificing lookup performance. We believe that file systems should adopt LSM tree techniques used by modern key-value stores to represent metadata and tiny files, because LSM trees aggressively aggregate metadata. Moreover, today's key-value store implementations are "thin" enough to provide the performance levels required by file systems. In our experiments, we used a LevelDB key-value store to implement TABLEFS.

TABLEFS uses modern key-value store techniques to pack small things (directory entries, inode attributes, small file data) into large on-disk files with the goal of suffering fewer seeks when seeks are unavoidable. It is a POSIX-compliant stacked file system that represents metadata and tiny files as key-value pairs using 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. By using stacking, TABLEFS asks for efficient large file allocation and access from the underlying local file system. By using an

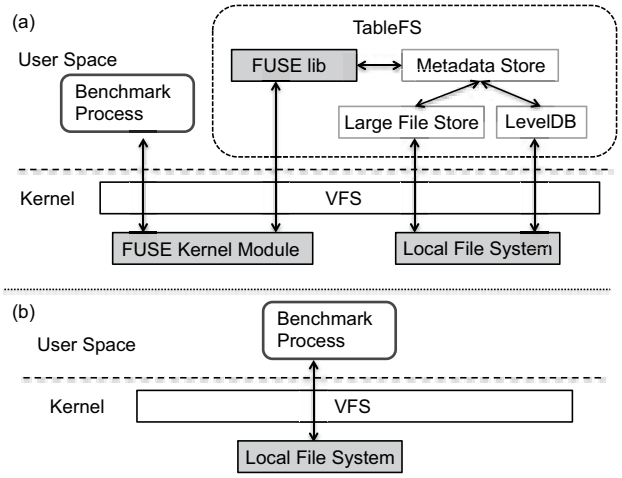


Figure 1: (a) The architecture of TABLEFS. A FUSE kernel module redirects file system calls from a benchmark process to TABLEFS, and TABLEFS stores objects into either LevelDB or a large file store. (b) When we benchmark a local file system, there is no FUSE overhead to be paid.

continued on page 11

FROM THE DIRECTOR'S CHAIR

Greg Ganger

Hello from fabulous Pittsburgh!

It has been another great year for the Parallel Data Lab. Some highlights include the return of database research, exciting new results on big pushes in cloud computing and "Big Data" systems, several prestigious awards, and continuing growth in PDL-related Masters programs. Along the way, many students graduated and joined PDL Consortium companies, new students joined the PDL, and many cool papers have been published. Let me highlight a few things.

I'll start with Andy Pavlo. It's been several years since we lost Natassa to Europe, and I'm thrilled to see Andy bringing back both database systems focus and, amazingly, a similar level of energy. He's a lot of fun and a great database systems researcher, and I (like others) am already enjoying working with him. You can see his background in the new faculty write-up about him… and, if you haven't seen him give a talk yet, you're in for a treat when you do.

As I noted last year, 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. I just wish we had coined either term, since PDL was active in both areas long before the buzzwords arose. Oh well. We continue to explore cool new systems approaches for supporting large-scale machine learning (a primary component of Big Data analytics), expand Masters program activities in both areas, and lead cloud computing research of the 6-institution Intel Science and Technology Center for Cloud Computing (ISTC-CC).

On the education front, we continue to expand our efforts to provide Masters students with excellent foundations in storage systems, cloud technologies, and Big Data systems. The storage systems class that Garth and I have taught for over 10 years had 100 students this year, and five excellent corporate guest lecturers (thank you, PDL Consortium members!). We also created a new cloud computing class, together with PDL alum Dr. Raja Sambasivan and Prof. Majd Sakr. Both classes serve several Masters programs, including the Masters program on data science systems that Garth has developed. That latter trains students with strong practical skills in the creation and exploitation of systems for Big Data analytics, including allowing 7-month internships to satisfy the program's capstone project requirement -- something in which many PDL companies may be interested in participating.

Several of us continue to work closely with Carnegie Mellon's excellent machine learning faculty to explore new systems for Big Data analytics. While the Map-Reduce approach is good for very simple data processing tasks, it is a poor tool for many of the advanced machine learning techniques that give "Big Data" its great promise. The front-page article describes one of the new approaches we've been exploring, and a number of others are emerging from our active brainstorming and exploration. Such cross-domain collaboration, which is a hallmark of Carnegie Mellon and PDL, is critical to the success of data sciences in practice.

Experiences like these have underscored our long-held belief that no single programming system is going to serve the breadth of data analytics styles and activities. Combining such systems with the breadth of other cloud computing activities, such as long-running services and others, leads to challenging resource scheduling challenges. For example, our Tetrisched project is developing new ways of allowing users to express their per-job resource type preferences (e.g., machine locality or hardware accelerators) and then exploring the trade-offs among them to maximize utility of the public and/or private cloud infrastruc-

THE PDL PACKET

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

PUBLISHER
Greg Ganger

EDITOR
Joan Digney

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

THE PDL LOGO

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

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

FACULTY

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

David Andersen
Lujo Bauer
Chuck Cranor
Lorrie Cranor
Christos Faloutsos
Eugene Fink
Rajeev Gandhi
Garth Gibson
Seth Copen Goldstein
Mor Harchol-Balter
Bruce Krogh
Todd Mowry
Onur Mutlu
Priya Narasimhan
David O'Hallaron
Andy Pavlo
Majd Sakr
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
Chad Dougherty
Zisimos Economou
Mitch Franzos
Charlene Zang

VISITING RESEARCHERS / POST DOCS

Samira Khan
Rolando Martins
Raja Sambasivan
Sadahiro Sugimoto

GRADUATE STUDENTS

Joy Arulraj
Rachata Ausavarungnirun
Ben Blum
Lei Cao
Jim Cipar
Henggang Cui
Utsav Drolia
Hartaj Dugal
Bin Fan
Jian Fang
Jesse Haber-Kucharsky
Junchen Jiang
Aditya Jaltade
Amod Jaltade
Wesley Jin
Saurabh Arun Kadekodi
Anuj Kalia
Mike Kasick
Peter Klemperer
Elie Krevat
Yang Li
Hyeontaek Lim
Thomas Marshall
Michelle Mazurek
Justin Meza
Nathan Mickulicz
Iulian Moraru
Kai Ren
Wolfgang Richter
Vivek Seshadri
Ilari Shafer
Pratik Shah
Mukul Singh
Yu Su
Jiaqi Tan
Alexey Tumanov
Hui Wang
Tejas Wanjari
Jinliang Wei
Lin Xiao
Lianghong Xu
HanBin Yoon
Huanchen Zhang
Jie Zhang
Rui Zhang
Qing Zheng
Dong Zhou
Timothy Zhu

FROM THE DIRECTOR'S CHAIR

ture. As another example, we are also exploring how to make storage and other stateful services more elastic and agile, so that mixes of services and frameworks can more effectively share cloud resources.

Naturally, our long-standing focus on scalable storage continues strongly. A primary challenge is metadata scaling, and PDL researchers are exploring several approaches to dealing with scale along different dimensions. For example, huge directories of files with structures names are sometimes used to organize huge numbers of related files, and novel scalable directory structures like GIGA+ offer an intriguing solution that simultaneously addresses scale in the number of directories as well. We are also exploring cool new approaches to exploiting log-based storage to accommodate high rates of metadata updates.

We continue to explore ways of exploiting the exciting new underlying storage technologies, such as NVM and Flash SSDs, to improve systems. This is one of several areas where Andy is active, as are several of the rest of us, exploring new ways of designing database systems to fully exploit the features of NVM. On the other end of the storage hierarchy, shingled magnetic recording (SMR) is changing the way the disk works, and we are exploring a range of new interface styles from hiding the difference to fully exposing it and leaving it to the host to manage — we call that "caveat scriptor".

Many other ongoing PDL projects are also producing cool results. For example, we are developing new approaches to providing storage SLOs with a focus on tail latencies. We continue to identify compelling architectures for using NVM with DRAM as part of hybrid main memories, exploiting NVM's superior energy characteristics rather than only its persistence. We also continue to create superior indexing structures and mechanisms to enable very efficient key-value stores in Flash and NVM, both for general storage services and for Big Data systems. 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 logs and data from these systems has been invaluable to our research on problem diagnosis, 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.

2013 PDL Retreat attendees enjoying a walk through the Bedford Springs grounds. From L to R: Michelle Mazurek, Peter Klemperer, Carolyn Connor (LANL), Jeff Heller (NetApp), Manisha Jain (Google), and Annika Peterson.

YEAR IN REVIEW

May 2014

- 16th annual PDL Spring Visit Day.
- Michelle Mazurek defended her dissertation on "A Tag-Based, Logical Access-Control Framework for Personal File Sharing."
- Aapo Kyrola defended his dissertation on "Large-scale Graph Computation on Just a PC."
- Jesse Haber-Kucharsky will be interning this summer at Microsoft Research in Redmond with Richard Draves on the Cosmos team.

April 2014

- Computer Science Professor Christos Faloutsos gave a keynote address at the 23rd International World Wide Web Conference this week in Seoul, South Korea on "Large Graph Mining: Patterns, Cascades, Fraud Detection, and Algorithms."
- Onur Mutlu received a Microsoft Research Award.
- Onur Mutlu's student Hyoseung Kim received the best paper award at RTAS '14 for his paper "Bounding Memory Interference Delay in COTS-based Multi-Core Systems."
- Gennady Pekhimenko proposed his Ph.D. thesis research on "Effective Data Compression for Modern Memory Systems."
- Vivek Seshadri proposed his Ph.D. thesis research on "Hardware Support for Fast and Energy-efficient Bulk Data Movement and Computation."
- Michelle Mazurek has accepted a job for the fall as an assistant professor in computer science at the University of Maryland.

March 2014

- Greg Ganger received the 2014 Steven J. Fenves Award for Systems Research.
- Wolfgang Richter and his co-authors Canturk Isci (IBM Research), Jan Harkes and Benjamin Gilbert (CMU), Vasanth Bala (IBM Research), and Mahadev Satyanarayan (CMU) won the best paper award at IC2E '14 for their paper "Agentless Cloud-wide Streaming of Guest File System Updates."
- Hyeontaek Lim proposed his Ph.D. thesis research on "Resource-Efficient Data-Intensive System Designs for High Performance and Capacity."

February 2014

- Onur Mutlu received and IBM Faculty Award.
- Priya Narasimhan's Yinzcam was used at the Superbowl!
- Kai Ren proposed his Ph.D. thesis research on "Fast Storage for File System Metadata."
- Wolfgang Richter proposed his Ph.D. thesis research on "Agentless Cloud-wide Monitoring of Virtual Disk State."
- Michelle Mazurek presented "Toward Strong, Usable Access Control for Shared Distributed Data" at FAST '14 in Santa Clara, CA.
- Lianghong Xu presented "SpringFS: Bridging Agility and Performance in Elastic Distributed Storage" at FAST '14 in Santa Clara, CA.
- Samira Khan presented "Improving Cache Performance by Exploiting Read-Write Disparity" at HPCA '14, in the best papers session, in Orlando, FL.
- Kiryong Ha's and Mahadev Satyanarayanan's paper "QuiltView: Glass-Sourced Video for Google Maps Queries" was presented at HotMobile'14 in Santa Barbara, CA. The Quiltview demo received the Best Demo Award.

January 2013

- Andy Pavlo went on a multi-city speaking tour in California, presenting "Cache Rules Everything Around Me" at several universities and industry bases.

December 2013

- Jiří Šimša defended his dissertation on "Systematic and Scalable Testing of Concurrent Programs."
- Vivek Seshadri presented "RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization" at the 46th International Symposium on Microarchitecture (MICRO) in Davis, CA.
- Gennady Pekhimenko presented "Linearly Compressed Pages: A Low-Complexity, Low-Latency Main Memory Compression Framework" at MICRO in Davis, CA.
- Qirong Ho presented "More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server" at NIPS '13 in Lake Tahoe, NV.

November 2013

- Garth Gibson was elected an IEEE Fellow.
- Mike Kasick presented "Making Problem Diagnosis Work for Large-Scale, Production Storage Systems" at LISA '13 in Washington, DC.
- Iulian Moraru presented "There Is More Consensus in Egalitarian Parliaments" SOSP'13 in Farmington, PA.
- Iulian Moraru presented "Consistent, Durable, and Safe Memory Management for Byte-addressable Non Volatile Main Memory" at TRIOS: Conference on Timely Results in Operating Systems, held in conjunction with SOSP '13.
- Michelle Mazurek presented "Measuring Password Guessability for an Entire University" at CCS: ACM Conference on Computer and Communications Security in Berlin, Germany.

October 2013

- Bin Fan defended his dissertation on "Algorithmic Engineering Towards More Efficient Key-Value Systems."
- Raja Sambasivan presented his paper on "Visualizing Request-flow Comparison to Aid Performance Diagnosis in Distributed Systems" at InfoVis'13 in Atlanta, GA.

continued on page 10

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

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

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

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

Shingled Magnetic Recording: Areal Density Increase Requires New Data Management

Tim Feldman & Garth Gibson

USENIX ;login:, v 38, n 3, June 2013.

Shingled Magnetic Recording (SMR) is the next technology being deployed to increase areal density in hard disk drives (HDDs). The technology will provide the capacity growth spurt for the teens of the 21st century. SMR drives get that increased density by writing overlapping sectors, which means sectors cannot be written randomly without destroying the data in adjacent sectors. SMR drives can either maintain the current model for HDDs by performing data retention behind the scenes, or expose the underlying sector layout, so that file system developers can develop SMR-aware file systems.

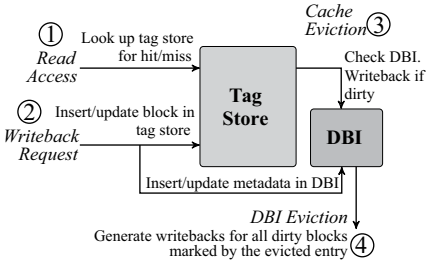
The Dirty-Block Index

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

41st International Symposium on Computer Architecture, June, 2014.

On-chip caches maintain multiple pieces of metadata about each cached block—e.g., dirty bit, coherence information, ECC. Traditionally, such metadata for each block is stored in the corresponding tag entry in the tag store. While this approach is simple to implement and scalable, it necessitates a full tag store lookup for any metadata query—resulting in high latency and energy consumption. We find that this approach is inefficient and inhibits several cache optimizations.

In this work, we propose a new way of organizing the dirty bit information that enables simpler and more efficient implementation of several optimizations. In our proposed approach, we remove the dirty bits from the tag store and organize it differently in a structure, which we call the Dirty-Block Index (DBI). The organization of DBI is simple: it consists of multiple entries, each corresponding to some row in DRAM. A bit vector in each entry tracks whether each block in the corresponding DRAM row is dirty or not.



Operation of a cache with DBI.

We demonstrate the effectiveness of DBI by using it to simultaneously implement three optimizations proposed by prior work: 1) Aggressive DRAM-aware writeback, 2) Bypassing cache lookups, and 3) Heterogenous ECC for clean/dirty blocks. DBI, with all three optimization enabled, improves performance by 31% compared to baseline (6% compared to the best previous mechanism) while reducing overall area cost by 8% compared to prior approaches.

Exploiting Bounded Staleness to Speed Up Big Data Analytics

Henggang Cui, James Cipar, Qirong Ho, Jin Kyu Kim, Seunghak Lee, Abhimanu Kumar, Gregory R. Ganger, Phillip B. Gibbons, Garth Gibson & Eric Xing

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-14-101. February 2014.

Many modern machine learning (ML) algorithms are iterative, converging on a final solution via many iterations over the input data. This paper explores approaches to exploiting these algorithms' convergent nature to improve performance, by allowing parallel and distributed threads to use loose consistency models for shared algorithm state. Specifically, we focus on bounded staleness, in which each thread can see a view of the current intermediate solution that may be a limited number of iterations out-of-date. Allowing staleness reduces communication costs (batched updates and cached reads) and synchronization (less waiting for locks or straggling threads). One approach is to increase the number of iterations between barriers in the oft-used Bulk Synchronous

continued on page 6

RECENT PUBLICATIONS

continued from page 5

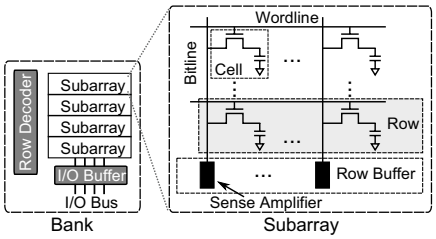
Parallel (BSP) model of parallelizing, which mitigates these costs when all threads proceed at the same speed. A more flexible approach, called Stale Synchronous Parallel (SSP), avoids barriers and allows threads to be a bounded number of iterations ahead of the current slowest thread. Extensive experiments with ML algorithms for topic modeling, collaborative filtering, and PageRank show that both approaches significantly increase convergence speeds, behaving similarly when there are no stragglers, but SSP outperforms BSP in the presence of stragglers.

Improving DRAM Performance by Parallelizing Refreshes with Accesses

Kevin Chang, Donghyuk Lee, Zeshan Chishti, Chris Wilkerson, Alaa Alameldeen, Yoongu Kim & Onur Mutlu

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

Modern DRAM cells are periodically refreshed to prevent data loss due to leakage. Commodity DDR (double data rate) DRAM refreshes cells at the rank level. This degrades performance significantly because it prevents an entire DRAM rank from serving memory requests while being refreshed. DRAM designed for mobile platforms, LPDDR (low power DDR) DRAM, supports an enhanced mode, called per-bank refresh, that refreshes cells at the bank level. This enables a bank to be accessed while another in the same rank is being refreshed, alleviating part of the negative performance impact of refreshes. Unfortunately, there are two shortcomings of per-bank refresh employed in today's systems. First, we observe that the per-bank refresh scheduling scheme does not exploit the full potential of overlapping refreshes with accesses across banks because it restricts the banks to be refreshed in a sequential round-robin order. Second, accesses to a bank that is being refreshed have to wait.



DRAM bank and subarray organization.

To mitigate the negative performance impact of DRAM refresh, we propose two complementary mechanisms, DARP (Dynamic Access Refresh Parallelization) and SARP (Subarray Access Refresh Parallelization). The goal is to address the drawbacks of per-bank refresh by building more efficient techniques to parallelize refreshes and accesses within DRAM. First, instead of issuing per-bank refreshes in a round-robin order, as it is done today, DARP issues per-bank refreshes to idle banks in an out-of-order manner. Furthermore, DARP proactively schedules refreshes during intervals when a batch of writes are draining to DRAM. Second, SARP exploits the existence of mostly-independent subarrays within a bank. With minor modifications to DRAM organization, it allows a bank to serve memory accesses to an idle subarray while another subarray is being refreshed. Extensive evaluations on a wide variety of workloads and systems show that our mechanisms improve system performance (and energy efficiency) compared to three state-of-the-art refresh policies and the performance benefit increases as DRAM density increases.

Agentless Cloud-wide Streaming of Guest File System Updates

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

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

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

Improving Cache Performance by Exploiting Read-Write Disparity

Samira Khan, Alaa Alameldeen, Chris Wilkerson, Onur Mutlu & Daniel Jimenez

Proceedings of the 20th International Symposium on High-Performance Computer Architecture (HPCA), Orlando, FL, February 2014. Best paper session.

Cache read misses stall the processor if there are no independent instructions to execute. In contrast, most cache write misses are off the critical path of execution, since writes can be buffered in the cache or the store buffer. With few exceptions, cache lines that serve loads are more critical for performance than cache lines that serve only stores. Unfortunately, traditional cache management mechanisms do not take into account this disparity between read-write criticality. The key contribution of this paper is the new idea of distinguishing between lines that are reused by reads versus those that are reused only by writes to focus cache management policies on the more critical read lines. We propose a Read-Write Partitioning (RWP) policy that minimizes read misses by dynamically partitioning the cache into clean and dirty partitions, where partitions

continued on page 7

continued from page 6

grow in size if they are more likely to receive future read requests. We show that exploiting the differences in read-write criticality provides better performance over prior cache management mechanisms. For a single-core system, RWP provides 5% average speedup across the entire SPEC CPU2006 suite, and 14% average speedup for cache-sensitive benchmarks, over the baseline LRU replacement policy.

We also show that RWP can perform within 3% of a new yet complex instruction-address-based technique, Read Reference Predictor (RRP), that bypasses cache lines which are unlikely to receive any read requests, while requiring only 5:4% of RRP's state overhead. On a 4-core system, our RWP mechanism improves system throughput by 6% over the baseline and outperforms three other state-of-the-art mechanisms we evaluate.

More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server

Qirong Ho, James Cipar, Henggang Cui, Jin Kyu Kim, Seunghak Lee, Phillip B. Gibbons, Garth A. Gibson, Gregory R. Ganger & Eric P. Xing

Conference on Neural Information Processing Systems (NIPS '13). Dec. 5-8, 2013, Lake Tahoe, NV.

We propose a parameter server system for distributed ML, which follows a Stale Synchronous Parallel (SSP) model of computation that maximizes the time computational workers spend doing useful work on ML algorithms, while still providing correctness guarantees. The parameter server provides an easy-to-use shared interface for read/write access to an ML model's values (parameters and variables), and the SSP model allows distributed workers to read older, stale versions of these values from a local cache, instead of waiting to get them from a central storage. This significantly increases the proportion of time workers spend computing, as opposed to waiting. Furthermore, the SSP model ensures ML algorithm correctness by limiting the maximum age of the stale values. We provide a proof of correctness under SSP, as well as empirical results demonstrating that the SSP model achieves faster algorithm convergence on several different ML problems, compared to fully-synchronous and asynchronous schemes.

QuiltView: Glass-Sourced Video for Google Maps Queries

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

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

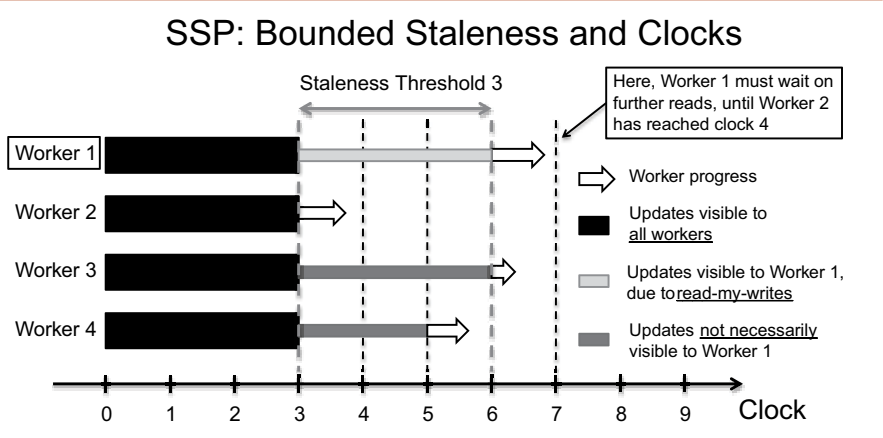
Effortless one-touch capture of video is a unique capability of wearable devices such as Google Glass. We use this capability to create a new type of crowd-sourced system in which users receive queries relevant to their current location and opt-in preferences. In response, they can send back live video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

Toward Strong, Usable Access Control for Shared Distributed Data

Michelle L. Mazurek, Yuan Liang, William Melicher, Manya Sleeper, Lujo Bauer, Gregory R. Ganger, Nitin Gupta & Michael K. Reiter

In FAST 2014: USENIX Conference on File and Storage Technologies, February 2014.

As non-expert users produce increasing amounts of personal digital data, usable access control becomes critical. Current approaches often fail, because they insufficiently protect data or confuse users about policy specification. This paper presents Penumbra, a distributed file system with access control designed to match users' mental models while providing principled security. Penumbra's design combines semantic, tag-based policy specification with logic-based access control, flexibly supporting intuitive policies while providing high assurance of correctness. It supports private tags, tag disagreement between users, decentralized policy enforcement, and unforgeable audit records. Penumbra's logic can express a variety of policies that map well to real users' needs. To evaluate Penumbra's design, we develop a set of detailed, realistic case studies drawn from prior research into users' access-control preferences. Using microbenchmarks and traces generated from the case studies, we demonstrate that Penumbra can



Bounded Staleness under the SSPModel.

continued on page 20

April 2014

Mutlu Receives Microsoft Research Award

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

-- Inside CIT

April 2014

RTAS Best Paper Award!

ECE Ph.D. student Hyoseung Kim won the Best Paper Award at the 2014 IEEE Real-time Technologies and Applications Symposium (RTAS) for a piece he co-authored with ECE Assistant Professor Onur Mutlu, ECE Professor Raj Rajkumar, and the SEI's Dionisio de Niz, Bjorn Andersson, and Mark Klein. The paper was titled "Bounding Memory Interference Delay in COTS-based Multi-Core Systems."

-- Inside CIT

March 2014

Greg Ganger Receives 2014 Steven J. Fenves Award

We are pleased to announce that Greg Ganger is the recipient of the 2014 Steven J. Fenves Award for Systems Research. He is the Jatras Professor of Electrical and Computer Engineering. He is also the director of the Parallel Data Lab.

The award is made to a CMU faculty member who has made a significant contribution to systems research in areas relevant to CMU's Institute for Complex Engineered Systems (ICES), through furthering the goal of interconnecting people, physical, and information, the development and demonstration of an engineered system, enhancing education in systems through the development of courses, publishing textbooks (paper or electronic), or a body of knowledge that is of pedagogical importance, or causing a paradigm shift in systems research. Greg is being recognized for his significant contributions to computer systems, in particular for his work on soft updates and self*-storage systems

-- with info from the ICES Newsroom, March 4, 2014

March 2014

IC2E Best Paper Award!

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

February 2014

Onur Mutlu Receives IBM Faculty Award

Assistant Professor Onur Mutlu has received a 2013 IBM Faculty Award in the amount of $40,000. This is the second year in a row that he has received this prestigious award. The IBM Faculty Award Program is a highly competitive program open to full-time faculty worldwide. Candidates must be nominated by an IBM employee who works in his or her research area and will then serve as liaison for their collaboration. According to the IBM website, winners of the IBM Faculty Award "must have an outstanding reputation for contributions in their field or, in the case of junior faculty, show unusual promise."

-- from ECE Online News February 21, 2014

February 2014

Lorrie Cranor International Science & Engineering Visualization Challenge People's Choice Award Winner

Lorrie Cranor, an associate professor of computer science and engineering and public policy, and director of the CyLab Usable Privacy and Security Laboratory, was recently recognized as one of 18 winners, honorable mentions and people's choice awardees from the International Science & Engineering Visualization Challenge. The contest, which is jointly run by the National Science Foundation (NSF) and the journal Science, exemplifies the old axiom, "a picture is worth a thousand wards." It celebrates the long tradition of using various types of illustrations to communicate the complexities of science, engineering and technology for education and journalistic purposes when words aren't enough. Cranor's quilt "Security Blanket" (pictured above) took honorable mention in the illustration category. While on sabbatical during the 2012-2013 academic

continued on page 9

continued from page 8

year she worked on visualizing security and privacy concepts through art as a fellow of the Frank-Ratchye STUDIO for Creative Inquiry.

-- 8.5x11 News, Feb. 6, Vol. 24, No. 28

February 2014

YinzCam Goes to the Superbowl!

Associate Professor of Electrical and Computer Engineering Priya Narasimhan, founder of YinzCam, and engineering students tested multi-camera instant replays to smartphones at the Super Bowl last Sunday. YinzCam, which creates mobile sports apps that allow fans to stay in touch with their favorite teams 24/7 by providing them with real-time stats, multimedia, streaming radio, social-media and more, has seen more than 7 million downloads of their products. The company's mobile-video technology is also being deployed within sports venues throughout the country to allow fans to watch instant replays, live cameras (including the NFL RedZone channel) on their smartphones, tablets or touchscreen computers.

-- 8.5x11 News, February 6, Vol. 24, No. 28

November 2013

Garth Gibson Elected IEEE Fellow

We are pleased to announce that Garth Gibson has been selected as an IEEE Fellow. Garth was selected for contributions to the performance and reliability of transformative storage systems. Becoming an IEEE Fellow is a distinction reserved for select IEEE members whose extraordinary accomplishments in any of the IEEE fields of interest are deemed fitting of this prestigious grade elevation.

September 2013

New Thinking Track at SDC'13

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

The PDL paper "LazyBase: Trading Freshness for Performance in a Scalable Database" by James Cipar, Greg Ganger, Kimberly Keeton, Charles B. Morrey III, Craig A. N. Soules, and Alistair Veitch was among the papers presented in this track. Originally published at Eurosys '12, it discusses scalable database system is specialized for the growing class of data analysis applications that extract knowledge from large, rapidly changing data sets.

July 2013

Support for Running Concurrent-write HPC Code on HDFS in PLFS

New code on running concurrent-write HPC codes on top of single-writer HDFS storage is now supported in the 2.4 release of PLFS. The code is available online at github.com/plfs.

July 2013

Call for Users: NSF PRObE 1000 Node Systems Research Testbed

Garth Gibson's long-running effort, in collaboration with Gary Grider of LANL and the New Mexico Consortium, to make a large-scale testbed available for systems researchers has come to fruition. Follow the link to find out how to apply for 1000 nodes for systems research experiments, via NSF's PRObE.

July 2013

Welcoming the Most Wonderfully Perfect Grandson in the World!

Congratulations to Karen Lindenfelser, who is the proud Grandma of Landon Thomas Ziants, born on July 8, 2013 to mom and dad Julie Lindenfelser and Charles Ziants. So much love has been added to the Lindenfelser family!

June 2013

Pavan Alampalli and Chinmay Kamat Receive Teaching Awards

Congratulations to Pavan and Chinmay on receiving their awards for Excellence as Teaching Assistants, specifically for their work with Garth and Greg on the Storage Systems class.

AWARDS & OTHER PDL NEWS

continued from page 4

- Onur Mutlu's papers "LightTx: A Lightweight Transactional Design in Flash-based SSDs to Support Flexible Transactions" and "Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation" were presented at ICCD '13 in Ashville, NC.
- 21st annual PDL Retreat.

August 2013

- Lin Xiao proposed her Ph.D. thesis research on "Scaling Metadata Service for Weak Scaling Workloads."
- Elie Krevat proposed his thesis research on "An Automated Approach for Mitigating Service Performance Problems with Efficient Resource Allocations."
- Onur Mutlu presented "Memory Scaling: A Systems Architecture Perspective" at MemCon 2013 (MEMCON) in Santa Clara, CA.
- Kai Ren presented "Hadoop's Adolescence: An Analysis of Hadoop Usage in Scientific Workloads" at VLDB '13 in Riva del Garda, Trento, Italy.

June 2013

- Pavan Alampalli and Chinmay Kamat received CMU teaching awards for their work on Garth's and Greg's storage systems class.
- Anshul Gandhi defended his dissertation on "Dynamic Server Provisioning for Data Center Power Management."
- Ilari Shafer presented "Specialized Storage for Big Numeric Time Series" at HotStorage '13 in San Jose, CA.
- Greg Ganger's paper "Active Disk Meets Flash: A Case for Intelligent SSDs" was presented at ICS'13 in Eugene, OR.
- Onur Mutlu's paper "An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms" was presented at ISCA '13 in Tel-Aviv, Israel.
- Justin Meza presented "A Case for Efficient Hardware/Software Cooperative Management of Storage and Memory" at WEED'13, held in conjunction with ISCA'13 in Tel-Aviv, Israel.
- Kai Ren presented "TABLEFS: Enhancing Metadata Efficiency in the Local File System" at the 2013 USENIX Annual Technical Conference in San Jose, CA.
- Kiryong Ha presented "Just-in-Time Provisioning for Cyber Foraging" at MobiSys'13 in Taipei, Taiwan.

May 2013

- Raja Sambasivan defended his dissertation on "Diagnosing Performance Changes in Distributed Systems by Comparing Request Flows."
- Bin Fu defended his dissertation on "Algorithms for Large-Scale Astronomical Problems."
- Soila Pertet Kavulya defended her dissertation on "Statistical Diagnosis of Chronic Problems in Production Systems."
- Peter Klemperer proposed his dissertation research on "Efficient Hypervisor Based Malware Detection.
- 15th annual PDL Spring Visit Day.

NEW PDL FACULTY

The PDL would like to welcome Andy Pavlo to its midst. Andy is an Assistant Professor at CMU in the Department of Computer Science. His research interests are in database management systems, specifically main memory systems, transaction processing systems (OLTP/NewSQL), non-relational systems (NoSQL), and large-scale data analytics (OLAP).

Andy is known as "the H-Store Guy," having spent a great deal of time as a grad student building the system. H-Store is an experimental main-memory, distributed database management system that is optimized for OLTP applications. It is a highly distributed, row-store-based relational database that runs on a cluster on shared-nothing, main memory executor nodes. He continues to collaborate on it with MIT, Brown University, QCRI, and Intel. Other interesting work he is collaborating on includes S-Store — an in-memory, distributed stream processing engine that is integrated with a front-end, OLTP database system; N-Store — a study of NVMs to understand their performance characteristics in the context of big data systems new DBMS architectures; and OLTP-Bench — an extensible "batteries included" DBMS benchmarking testbed with over a dozen workloads that all differ in complexity and system demands.

As with most American legends, it is difficult to discern what is true versus what is lore. Andy is a native Marylander whose strong wanderlust and belief in the Aristotlean supremacy of the pursuit of knowledge has taken him to places like R.I.T. in New York and the University of Wisconsin-Madison. He completed his PhD in 2013 at Brown University on the New England coast under the prudent guidance of Dr. Stanley Zdonik. He has raised award-winning clams and was a leading exponent of the womb-core music movement in the early 2000s. Andy truly is a man of the times.

continued from page 1

LSM tree, TableFS ensures metadata is written to disk in large, non-overwrite, sorted and indexed logs.

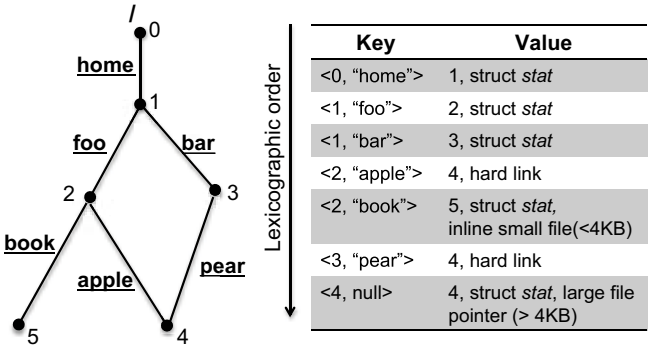
Promising tests indicate that even an inefficient FUSE based user-level implementation of TABLEFS can perform comparably to Ext4, XFS and Btrfs on data-intensive benchmarks, and can outperform them by 50% to as much as 1000% for metadata-intensive workloads.

TableFS

TableFS represents directories, inodes and small files in one all-encompassing table, and only writes large objects (such as write-ahead logs, sorted collections of key-value pairs called SSTables, and large files) to the local disk.

There is no explicit space management in TableFS. Instead, it uses an underlying large-file-optimized local file system for allocation and storage of objects. Because TableFS packs directories, inodes and small files into a LevelDB table, and LevelDB stores sorted logs (SSTables) of about 2MB each, the local file system sees many fewer, larger objects. Files larger than T bytes are stored directly in the object store named according to their inode number. The object store uses a two-level directory tree in the local file system, storing a file with inode number I as "/LargeFileStore/J/I" where $J = I \div 10000$. This is to circumvent any scalability limit on directory entries in the underlying local file systems.

Daniel Tennant (NetApp), Bill Courtright and Chuck Cranor (CMU) at the 2013 PDL Retreat at Bedford Springs Resort.



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

Figure 2: 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.

TableFS pursues an aggressive clustering strategy; each row of a table is ordered in the table with its parent directory, embedding directory entries, inode attributes and the data of small files. The clustering manifests as adjacency for objects in the lower level object store if these entries were created/updated close together in time, or after compaction has merge sorted them back together. To link together the hierarchical structure of the user's namespace, the rows of the table are ordered by a variable-length key consisting of a 64-bit inode number indicating the file's parent directory and its filename string. The value of a row contains inode attributes, such as inode number, file size and timestamps. For small files, the file's row also contains the file's data.

Figure 2 shows an example of storing a sample file system's metadata into one LevelDB table. All entries in the same directory have rows that share the same first 64 bits of their table key, so they are clustered together. For readdir operations, once the inode number of the target directory has been retrieved, a scan sequentially lists all entries having the directory's inode number as the first 64 bits of their table key. To resolve a single pathname, TableFS starts searching from the root inode, which has a well-known inode number (0). Traversing the user's directory tree involves constructing a search key by concatenating the inode number of current directory with the hash of next component name in the pathname.

TableFS utilizes the scan operation provided by LevelDB to implement readdir() system call. The scan operation in LevelDB is designed to support iteration over arbitrary key ranges, which may require searching multiple SSTables. In such a case, Bloom filters cannot help to reduce the number of SSTables to search. However, in TableFS, readdir() only scans keys sharing the common prefix — the inode number of the searched directory. For each SSTable, an additional Bloom filter is maintained, to keep track of all inode numbers that appear as the first 64 bit of row keys in the SSTable. Before starting an iterator in an SSTable for readdir(), TableFS can first check its Bloom filter to find out whether it contains any of the desired directory entries. Therefore, unnecessary iterations over SSTables that do not contain any of the requested directory entries can be avoided.

LevelDB provides atomic insertion of a batch of writes but does not support atomic row read-modify-write operations. The atomic batch write guarantees that a sequence of updates

continued on page 12

continued from page 11

to the database are applied in order, and committed to the write-ahead log atomically. Thus the rename operation can be implemented as a batch of two operations: insert the new directory entry and delete the stale entry. But for operations like chmod and utime, since all of an inode's attributes are stored in a single key-value pair, TableFS must read-modify-write attributes atomically. We implemented a light-weight locking mechanism in the TableFS core layer to ensure correctness under concurrent access.

Journaling for TableFS relies on LevelDB and the local file system. LevelDB can be set to commit the log to disk synchronously or asynchronously; by default it commits the write-ahead log to disk every 5 seconds to match the semantics of the way most local file systems are used.

Evaluating TableFS

We evaluate our TableFS prototype on Linux (Ubuntu 12.10, Kernel 3.6.6 64-bit version) desktops equipped with an AMD 242 Dual Core Opteron Processor with 16GB DDR SDRAM

Peter Klemperer discusses his research on "Efficient Hypervisor Based Introspection with Snapshots" with Jeff Heller (NetApp).

running on a Western Digital 2T, 7200 RPM SATA (random seeks - 100 seeks/sec peak, sequential reads - 137.6 MB/sec peak, sequential writes 135.4 MB/sec peak).

For our analysis, we imagine three TableFS configurations: A kernel-native TableFS – a stacked file system, where the second local file system is treated as an object store (Figure 3(a)); a FUSE-based user-level TableFS (Figure 3(b)) with no TableFS function in the kernel and all of TableFS in the user level FUSE daemon); and an application-embedded TableFS library (Figure 3(c)). We implement the latter two to bracket the performance of the former.

We compare TableFS in its various configurations with Linux's most sophisticated local file systems: Ext4, XFS, and Btrfs. Ext4 is mounted with "ordered" journaling to force all data to be flushed out to disk before its metadata is committed to disk. By default, Ext4's journal is asynchronously committed to disks every 5 seconds. XFS and Btrfs use similar policies to asynchronously update journals. Since the tested filesystems have different inode sizes, we pessimistically penalize TableFS by padding its inode attributes to 256 bytes. This slows down TableFS doing metadata-intensive workloads significantly, but it still performs quite well. In some benchmarks, the Linux boot parameters were also changed to limit the machines' available memory below a certain threshold, in order to out-of-RAM performance.

Following is an example of the tests we ran to evaluate TableFS. For a more in-depth look at the ways we assessed the performance of TableFS, please see Ren [ATC'13]. In addition to the experiment outlined below, we examined the cost of FUSE overhead in TableFS, the performance of an in-kernel TableFS, and TableFS performance without FUSE overhead. We also tested the performance of TableFS using readdir operations, and did benchmark testing on directories as 100 million files were created – this latter test shows TableFS at its best, doing efficient batching of creation records, where it achieves up to 10X faster create rates, shown in Figure 4.

We also run two sets of macrobenchmarks on the FUSE version of TableFS, which provides a full featured, transparent application ser-

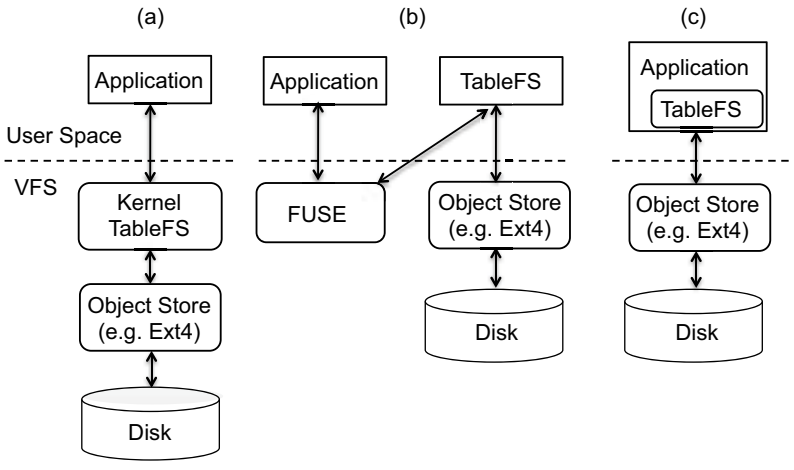


Figure 3: Three different implementations of TABLEFS: (a) the kernel-native TABLEFS, (b) the FUSE version of TABLEFS, and (c) the library version of TABLEFS. In the following evaluation section, (b) and (c) are presented to bracket the performance of (a), which was not implemented.

continued on page 13

continued from page 12

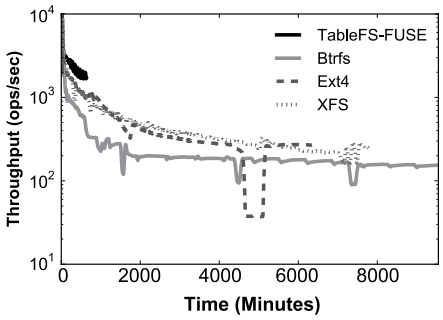


Figure 4: Throughput of all four tested file systems while creating 100 million zero-length files. TABLEFS-FUSE is almost 10X faster than the other tested file systems in the later stage of this experiment. The data is sampled in every 10 seconds and smoothed over 100 seconds. The vertical axis is shown on a log scale.

vice. Our goal is to demonstrate that TableFS is capable of reasonable performance for the traditional workloads that are often used to test local file systems.

Kernel build is a macrobenchmark that uses a Linux kernel compilation and related operations to compare TableFS performance to the other tested file systems. In the kernel build test, we use the Linux 3.0.1 source tree (whose compressed tar archive is about 73 MB in size). In this test, we run four operations in this order: untar the source tarball; grep "nonexistent pattern" over all of the source tree; run make inside the source tree; and gzip the entire source tree. After compilation, the source tree contains 45,567 files with a total size of 551MB. The machine's available memory is set to be 350MB, and therefore compilation data is forced to be written to the disk. Summing the operations, TableFS-FUSE is about 20% slower, but it is paying significant overhead caused by moving all data through the user-level FUSE daemon and the kernel twice, rather than once, as illustrated in Figure 3(b). The performance of Ext4, XFS, and Btrfs also degrades similarly when accessed through FUSE.

Postmark was designed to measure the performance of a file system used for e-mail, and web based services. It creates a large number of small randomly sized files between 512B and 4KB, performs a specified number of transactions on them, and then deletes all of them. Each transaction consists of two sub-transactions, with one being a creation or delete and the other being a read or append. The configuration used for these experiments consists of two million transactions on one million files, and the biases for transaction types are equal. The experiments were run with the available memory set to be 1400 MB, too small to fit the entire datasets (about 3GB) in memory.

For Postmark, TableFS outperforms other tested file systems by at least 23% during the transactions phase. TableFS runs faster than the other tested file systems for read, append and deletion, but runs slower for the creation. In Postmark, the creation phase creates files in the alphabetical order of their filenames. Thus it is a sequential insertion workload for all file systems, and Ext4 and XFS perform very efficiently in this workload. Again, TableFS-FUSE pays for the overhead from FUSE and writing file data at least twice: LevelDB first time writes it to the write-ahead log, and second time to an SSTable during compaction.

Conclusion

Our implementation of TableFS, which uses modern key-value store techniques to pack small things (directory entries, inode attributes, small file data) into large on-disk files with the goal of suffering fewer seeks when seeks are unavoidable, even hampered by FUSE overhead, LevelDB code overhead, LevelDB compaction overhead, and pessimistically padded inode attributes, performs as much as 10 times better than state-of-the-art local file systems in extensive metadata update workloads.

The lively entertainment at Bedford Springs.

Our current work with TableFS merges it with GIGA+, a directory partitioning scheme (Patil [FAST'11]), to make a stacked, scalable file system that can be layered on top of large-file-optimized distributed file systems and enable fast metadata and small file performance to scale to many metadata servers. We are applying this to high performance parallel file system and to cloud file systems like HDFS.

References

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

[ATC'13] TableFS: Enhancing Metadata Efficiency in the Local File System. Kai Ren, Garth Gibson. 2013 USENIX Annual Technical Conference, June 26-28, 2013, San Jose, CA.

DISSERTATION ABSTRACT:
Large-scale Graph Computation on Just a PC

Aapo Kyrola

Carnegie Mellon University SCS

Ph.D. Dissertation, May 7, 2014

Current systems for graph computation require a distributed computing cluster to handle very large real-world problems, such as analysis on social networks or the web graph. While distributed computational resources have become more accessible, developing distributed graph algorithms still remains challenging, especially to non-experts.

In this work, we present GraphChi, a disk-based system for computing efficiently on graphs with billions of edges. By using a well-known method to break large graphs into small parts, and a novel Parallel Sliding Windows algorithm, GraphChi is able to execute several advanced data mining, graph mining, and machine learning algorithms on very large graphs, using just a single consumer-level computer. We show, through experiments and theoretical analysis, that GraphChi performs well on both SSDs and rotational hard drives.

We build on the basis of Parallel Sliding Windows to propose a new data structure, the Partitioned Adjacency Lists, which we use to design an online graph database, GraphChi-DB. We demonstrate that, on a single PC, GraphChi-DB can process over one hundred thousand graph updates per second, while simultaneously performing computation. GraphChi-DB compares favorable to existing graph databases, particularly on data that is much larger than the available memory.

We evaluate our work both experimentally and theoretically. Based on the Parallel Sliding Windows algorithm, we propose new I/O efficient algorithms for solving fundamental graph problems. We also propose a novel algorithm for simulating billions of random walks in parallel on a single computer.

By repeating experiments reported for existing distributed systems, we show that, with only fraction of the resources, GraphChi can solve the same problems in very reasonable time. Our work makes large-scale graph computation available to anyone with a modern PC.

Gennady Pekhimenko describes his research on "Linearly Compressed Pages: A Low-Complexity, Low-Latency Main Memory Compression Framework" to Jerry Fredin (NetApp).

DISSERTATION ABSTRACT:
A Tag-Based, Logical Access-Control Framework for Personal File Sharing

Michelle Mazurek, ECE

Ph.D. Dissertation, May 6 2014

People store and share ever-increasing numbers of digital documents, photos, and other files, both on personal devices and within online networks. In this environment, proper access control is critical to help users obtain the benefits of sharing varied content with different groups of people while avoiding trouble at work, embarrassment, identity theft, and other problems related to unintended disclosure. Current approaches often fail, either because they insufficiently protect data or because they confuse users about policy specification. Historically, correctly managing access control has proven difficult, time-consuming, and error-prone, even for experts; to make matters worse, access control remains a secondary task most non-expert users are unwilling to spend significant time on.

To solve this problem, access control for file-sharing tools and services should provide verifiable security, make policy configuration and management simple and understandable for users, reduce the risk of user error, and minimize the required user effort. This thesis presents three user studies that provide insight into people's access-control needs and preferences. Drawing on the results of these studies, I present Penumbra, a prototype distributed file system that combines semantic, tag-based policy specification with logic-based access control, flexibly supporting intuitive policies while providing high assurance of correctness. Penumbra is evaluated using a set of detailed, realistic case studies drawn from the presented user studies. Using microbenchmarks and traces generated from the case studies, Penumbra can enforce users' policies with overhead less than 5% for most system calls. Finally, I present lessons learned, which can inform the further development of usable access-control mechanisms both for sharing files and in the broader context of personal data.

DISSERTATION ABSTRACT:
Systematic and Scalable Testing of Concurrent Programs

Jiří Šimša

Carnegie Mellon University SCS

Ph.D. Dissertation, December 16, 2013

The challenge this thesis addresses is to speed up the development of concurrent programs by increasing the efficiency with which concurrent programs can be tested and consequently evolved. The goal of this thesis is to generate methods and tools that help software engineers increase confidence in the correct operation of their programs. To achieve this goal, this thesis advocates testing of concurrent software using a systematic approach

continued on page 15

continued from page 14

capable of enumerating possible executions of a concurrent program.

The practicality of the systematic testing approach is demonstrated by presenting a novel software infrastructure that repeatedly executes a program test, controlling the order in which concurrent events happen so that different behaviors can be explored across different test executions. By doing so, systematic testing circumvents the limitations of stochastic testing, which relies on chance to discover concurrency errors.

However, the idea of systematic testing alone does not quite solve the problem of concurrent software testing. The combinatorial nature of the number of ways in which concurrent events of a program can execute causes an explosion of the number of possible interleavings of these events, a problem referred to as state space explosion.

To address the state space explosion problem, this thesis studies techniques for quantifying the extent of state space explosion and explores several directions for mitigating state space explosion: parallel state space exploration, restricted runtime scheduling, and abstraction reduction. In the course of its research exploration, this thesis pushes the practical limits of systematic testing by orders of magnitude, scaling systematic testing to real-world programs of unprecedented complexity.

DISSERTATION ABSTRACT:
Algorithmic Engineering Towards More Efficient Key-Value Systems

Bin Fan

Carnegie Mellon University SCS

Ph.D. Dissertation, October 24, 2013

Distributed key-value systems have been widely used as elemental components of many Internet-scale services at sites such as Amazon, Facebook and Twitter. This thesis examines a system design approach to scale existing key-value systems, both horizontally and vertically, by carefully engineering and integrating

Enjoying the sunshine at Bedford Springs. From L to R: Jerry Fredin (NetApp), Nitin Agrawal (NEC), Jorge Guerra (VMware), and Abhijit Paithankar (VMware).

techniques that are grounded in recent theory but also informed by underlying architectures and expected workloads in practice. As a case study, we redesign FAWN-KV (i.e., a distributed key-value cluster consisting of wimpy key-value nodes) to achieve higher memory efficiency and ensure higher throughput even in the worst case.

First, to improve the worst-case throughput of a FAWN-KV system, we propose a randomized load balancing scheme that can fully utilize all the nodes regardless of their query distribution. We analytically prove and empirically demonstrate that deploying a very small but extremely fast load balancer at FAWN-KV can effectively prevent uneven or dynamic workloads creating hotspots on individual nodes. Moreover, our analysis provides service designers a mathematically tractable approach to estimate the worst-case throughput and help them avoid drastic over-provisioning in similar distributed key-value systems.

Second, to implement the extremely high-speed load balancer and also to improve the space efficiency of individual key-value nodes, we propose novel data structures and algorithms, including cuckoo filter, a Bloom filter replacement that is high-speed, highly compact and delete-supporting, and optimistic cuckoo hashing, a fast and space-efficient hashing scheme that scales on multiple CPUs. Both algorithms are built upon conventional cuckoo hashing but are optimized for our target architectures and workloads. Using them as building blocks, we design and implement MemC3 to serve transient data from DRAM with high throughput and low-latency retrieval, and SILT to provide cost-effective access to persistent data on flash storage with extremely small memory footprint (e.g., 0.7 bytes per entry).

DISSERTATION ABSTRACT:
Dynamic Server Provisioning for Data Center Power Management

Anshul Gandhi

Carnegie Mellon University SCS

Ph.D. Dissertation, June 2013

Data centers play an important role in today's IT infrastructure. However, their enormous power consumption makes them very expensive to operate. Sadly, much of the power used by data centers is wasted because of poor capacity management, leading to low server utilization.

In order to reduce data center power consumption, researchers have proposed several dynamic server provisioning approaches. However, there are many challenges that hinder the successful deployment of dynamic server provisioning, including: (i) unpredictability in workload demand, (ii) switching costs when setting up new servers, and (iii) unavailability of data when provisioning stateful servers. Most of the existing research in dynamic server provisioning has ignored, or carefully sidestepped, these important challenges at the expense of reduced benefits. In order to realize the full potential of dynamic server provisioning, we must overcome these associated challenges.

This thesis provides new research contributions that explicitly address the open challenges in dynamic server provisioning. We fist develop novel performance modeling tools to estimate the effect of these challenges on

continued on page 16

continued from page 15

response time and power. In doing so, we also address several long-standing open questions in queueing theory, such as the analysis of multi-server systems with switching costs. We then present practical dynamic provisioning solutions for multi-tier data centers, including novel solutions that allow scaling the stateful caching tier and solutions that are robust to load spikes. Our implementation results using realistic workloads and request traces on a 38-server multi-tier testbed demonstrate that dynamic server provisioning can successfully meet typical response time guarantees while significantly lowering power consumption.

While this thesis focuses on server provisioning for reducing power in data centers, the ideas presented herein can also be applied to: (i) private clouds, where unneeded servers can be repurposed for "valley-filling" via batch jobs, to increase server utilization, (ii) community clouds, where unneeded servers can be given away to other groups, to increase the total throughput, and (iii) public clouds, where unneeded virtual machines can be released back to the cloud, to reduce rental costs.

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

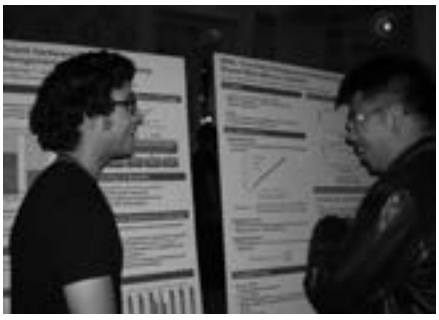
Carnegie Mellon University ECE
Ph.D. Dissertation, May 1, 2013

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 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.

continued on page 17



Justine Meza discusses "A Case for Efficient Hardware/Software Cooperative Management of Storage and Memory" with Yang Seok Ki (Samsung) at the 2013 PDL Retreat.

continued from page 16

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.

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.

THESIS PROPOSAL:
Effective Data Compression for Modern Memory Systems

Gennady Pekhimenko, SCS

April 22, 2014

The recent Big Data revolution challenges existing computer systems to perform computation near massive data sets. Modern memory systems have to address this challenge by providing both sufficient capacity across multiple layers of the memory hierarchy (including caches, DRAM, and non-volatile memory technologies) as well as sufficient bandwidth interconnect (including off-chip and on-chip buses) to transfer data between these layers. However, because DRAM and caches already constitute a significant portion of the system's cost and power budget, adding more DRAM and/or caches to meet increasing application demands is not desirable.

In this proposal, I present a series of mechanisms that exploit the existing redundancy in applications' data to perform efficient compression in caches and main memory, thereby providing higher effective capacity and higher available bandwidth across the memory hierarchy. To make data compression practical for both on-chip caches and main memory, in our prior work, we designed and implemented both an efficient hardware data compression algorithm (Base-Delta-Immediate compression), as well as a main memory compression framework (Linearly Compressed Pages) that can exploit different compression algorithms to provide significant increases in both capacity and available off-chip bandwidth.

This thesis proposal extends these preliminary steps in several major directions. First, we aim to design and evaluate compression-aware management policies that take into account compressed cache block size along with temporal locality to improve the performance of compressed caches. Second, we will investigate software-based transformations that can potentially improve applications' data compressibility. Third, we aim to apply our techniques to different computational platforms (e.g., GPUs) and new emerging applications (e.g., visual computing workloads) that can significantly benefit from the additional bandwidth provided by data compression.

THESIS PROPOSAL:
Hardware Support for Fast and Energy-efficient Bulk Data Movement and Computation

Vivek Seshadri, SCS

April 3, 2014

With growing compute power of modern multi-core systems and with the increasing amount of data accessed by many applications, the memory channel will become a even more critical bottleneck for both system performance and energy-efficiency. Existing systems use memory as just a data store in which data can be stored and accessed from. However, this approach necessitates large amounts of data to be transferred over the memory channel even for some simple operations like bulk data copy or initialization. This results in high latency, bandwidth and energy consumption for such operations.

This thesis presents a series of mechanisms that will allow the processor to

continued on page 18

continued from page 17

perform certain bulk data operations completely within DRAM, thereby eliminating the need to transfer large amounts of data over the memory channel. To keep the cost of DRAM low, our mechanisms aim to exploit the organization and operation of DRAM as much as possible. As a preliminary step, we have already developed and evaluated a mechanism (RowClone) for performing bulk data copy and initialization completely within DRAM.

This thesis proposal consists of two parts. The first part aims to explore other bulk data operations (e.g., gather-scatter, randomization) to DRAM. The second part aims to improve the hardware support for bulk data copy and initialization, further improving performance and energy efficiency compared to RowClone. If successful, we expect the mechanisms proposed by this thesis to significantly advance the state-of-the-art in many important applications.

THESIS PROPOSAL:
Resource-Efficient Data-Intensive System Designs for High Performance and Capacity

Hyeontaek Lim, SCS

March 24, 2014

The key contributions of my thesis are algorithms and data structures for storing and processing a large amount of fine-grained data on modern hardware, and software architectures that combine and tune these technologies together to build data-intensive systems that achieve high performance and use memory efficiently. This thesis will describe SILT and MICA, which are key-value stores providing a hash table-like interface, as examples that demonstrate how these algorithms, data structures, and software architectures apply to data-intensive system designs. SILT, which is based upon Entropy-Coded Tries that index items sorted by the hash of their keys, requires only 0.7 bytes per item in memory, serving requests at flash drive speeds of tens to hundreds of thousands of items per second. MICA, which uses lossy and lossless hash indexes, circular logs, an client-assisted hardware-based request direction, handles 65.6 million remote operations per second per server node for items stored in memory. My proposed work will apply an additional set of algorithms and data structures to strengthen SILT and MICA's benefits and improve their robustness on diverse hardware.

Alexey Tumanov talks about his research on "Tetrisched: Space-Time Scheduling for Heterogeneous Datacenters" at the 2013 PDL Retreat.

THESIS PROPOSAL:
Fast Storage for File System Metadata

Kai Ren, SCS

February 12, 2014

In an era of big data, the rapid growth of data that many companies and organizations produce and manage continues to drive efforts to improve the scalability of storage systems. The number of objects presented in storage systems continue to grow, making metadata management critical to the overall performance of file systems. On the other hand, many modern parallel applications are shifting toward shorter durations and larger degree of parallelism. Such trends continue to make storage systems to experience more diverse metadata intensive workloads.

The goal of this dissertation is to improve metadata management in both local and distributed file systems. The dissertation focuses on two aspects. One is to improve the out-of-core representation of file system metadata, by exploring the use of log-structured multi-level approaches to provide a unified and efficient representation in versatile secondary storage devices (e.g., traditional hard disk, shingled disk, and solid state disk). The other aspect is to demonstrate that such representation also can be flexibly integrated with many namespace distribution mechanisms to scale metadata performance of distribution file systems, and provide better support for big data analytic applications in data center environment.

THESIS PROPOSAL:
Agentless Cloud-wide Monitoring of Virtual Disk State

Wolfgang Richter, SCS

February 6, 2014

This dissertation proposes a fundamentally different way of monitoring virtual disk state in the cloud. The proposed platform is both agentless—meaning it operates external to and independent of the virtual servers it monitors—and scalable—meaning it is designed to efficiently address collections of virtual servers numbering in the thousands. The core technology used to create this platform is called Distributed Streaming Virtual Machine Introspection (DS-VMI), and it leverages two properties of modern clouds: virtualized servers managed by Virtual Machine Monitors (VMMs) enabling efficient introspection, and file-level duplication of data within cloud instances.

We explore a new class of agentless monitoring applications via three interfaces with two different consistency models: cloud-inotify (strong consistency), /cloud (eventual consistency), and /cloud-history (strong consistency). cloud-inotify is a pub-

continued on page 19

continued from page 18

lish-subscribe interface to cloud-wide file-level updates and it supports event-based monitoring applications. /cloud is designed to support batch-based and legacy monitoring applications by providing a file system interface to cloud-wide file-level state. /cloud-history is designed to support efficient search and management of historic virtual disk state. Achieving distributed near-real-time file-level deduplication, key for scalability, leads to a novel application of an incremental hashing construction. We also describe a novel snapshotting method combining the properties of both black box and white box methods which creates near-real-time file-level deduplicated snapshots of virtual disks.

THESIS PROPOSAL:
An Automated Approach for Mitigating Service Performance Problems with Efficient Resource Allocations

Elie Krevat, ECE

August 23, 2013

Distributed and cloud computing services are increasingly built atop a preexisting infrastructure of shared services. These services have separate performance characteristics and require enough resources to support each application's service level objectives (SLOs), while preferably not wasting too many resources from overprovisioning. Changes in a service's performance are common (e.g., multiple times per day) for any number of reasons, such as from modified system configurations, hardware failures, or increased loads. Even worse, a problem in any one service can cause cascading delays across a complex web of interdependent services.

In this proposal, we describe an automated approach to mitigating such performance problems through reactive resource provisioning. When a problem occurs, we attempt to mitigate the problem in the short term by automatically assigning the right types and quantities of resources across services that can usefully apply them. Our proposed approach makes use of end-to-end request traces to determine the actual service flow and synchronicity requirements, combined with resource usage statistics to determine specific demands. This general monitoring framework is also used to discover each service's elastic scaling properties and to provide online feedback to better evaluate resource assignments.

THESIS PROPOSAL:
Efficient Hypervisor Based Malware Detection

Peter Friedrich Klemperer, ECE

May 28, 2013

Recent years have seen an uptick in master boot record (MBR) based rootkits that load before the Windows operating system and subvert the operating system's own procedures. As such, MBR rootkits are difficult to counter with operating system-based antivirus software that runs at the same privilege-level as the rookits. Hypervisors operate at a higher privilege level than the guests they manage, creating a high-ground position in the host. This high-ground position can be exploited to perform security checks on the virtual machine guests where the checking software is isolated from guest-based viruses. The system proposed in this prospectus will target existing hypervisor systems to improve security with real-time, coherent memory introspection capabilities.

High performance guest memory introspection will decouple memory introspection from virtual machine guest execution, establish coherent and consistent memory views between the host and running guest, and provide intelligent memory translation to accelerate host-to-guest memory access. Existing introspection systems have provided one or two of these properties but not all three at once. This prospectus will present a new concurrent-computing approach to accelerate hypervisor based introspection of virtual machine guest memory that combines all three elements to improve performance and security.

The proposed system accelerates existing introspection systems and enables security protection techniques previously dismissed as too slow. In this prospectus, I will explain why existing introspection systems are inadequate, show how existing system performance can be improved, plan an initial prototype, and present several demonstrating applications based on that prototype. These demonstrating applications will be used to evaluate the prototype's performance and utility for supporting security applications.

THESIS PROPOSAL:
Scaling Metadata Service for Weak Scaling Workloads

Lin Xiao, SCS

August 5, 2013

Many file systems achieve high performance and scalability for large files instead of large number of files by striping or chunking files into data servers. However, as observed in real clusters, small files dominate the namespace, so metadata service could become the bottleneck as systems grow. The thesis is attacking the metadata scaling problem for weak scaling workloads. Metadata workloads in which file metadata operations increased much faster than directory name operations are defined as weak scaling workloads.

In this proposal, we present ShardFS, a hybrid solution to replicate directory names and sharding file metadata on oblivious metadata servers with demonstration using Hadoop file system(HDFS). We show the correctness and scalability of the system. We also focus on how to store metadata on disk to achieve similar performance as when they fit in memory. Finally we discuss the trade-offs on building metadata service with scalable distributed B-tree.

continued from page 7

enforce users' policies with overhead less than 5% for most system calls.

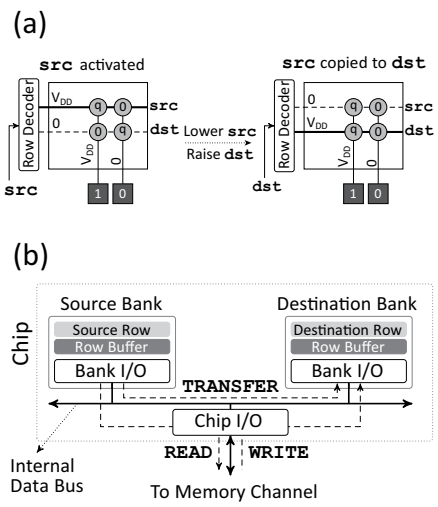
RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization

Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee, Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo, Onur Mutlu, Michael A. Kozuch, Phillip B. Gibbons & Todd C. Mowry

Proceedings of the 46th International Symposium on Microarchitecture (MICRO), Davis, CA, December 2013.

Several system level operations trigger bulk data copy or initialization. Despite the fact that these bulk data operations do not require any computation, current systems transfer a large quantity of data back and forth on the memory channel to perform such operations. As a result, bulk data operations consume high latency, bandwidth, and energy — degrading both system performance and energy efficiency.

In this work, we propose RowClone, a new and simple mechanism to perform bulk copy and initialization operations completely within DRAM — eliminating the need to transfer any data over the memory channel to perform such operations. Our key observation is that DRAM can internally and efficiently transfer a large quantity of data (multiple KBs) between a row of DRAM cells and the associated row-buffer. Based on this, our primary mechanism can copy an entire row's worth of data between two rows that share a row-buffer. This mechanism, which we call the Fast Parallel Mode, can reduce the latency and energy of a bulk copy operation by 11.6x and 74.4x, respectively. To efficiently copy data across rows that do not share a row-buffer, we propose a second mode of RowClone, the Pipelined Serial Mode. RowClone requires only a 0.01% increase in DRAM chip area. We quantitatively evaluate the



Fast Parallel Mode of RowClone (a) and Pipelined Serial Mode of RowClone (b).

benefits of RowClone using fork, one of the most frequently invoked system calls, and five copy and initialization intensive applications/phases. Our results show that RowClone can significantly improve both single-core and multi-core system performance, while also significantly reducing energy consumption.

Linearly Compressed Pages: A Low-Complexity, Low-Latency Main Memory Compression Framework

Gennady Pekhimenko, Vivek Seshadri, Yoongu Kim, Hongyi Xin, Onur Mutlu, Michael A. Kozuch, Phillip B. Gibbons & Todd C. Mowry

Proceedings of the 46th International Symposium on Microarchitecture (MICRO), Davis, CA, December 2013.

Data compression is a promising approach for meeting the increasing memory capacity demands expected in future systems. Unfortunately, existing compression algorithms do not translate well when directly applied to main memory because they require the memory controller to perform non-trivial computations to locate a cache line within a compressed memory page, thereby increasing access latency and degrading system performance. Prior proposals for addressing this performance degradation problem are either costly or energy inefficient.

By leveraging the key insight that all cache lines within a page should be compressed to the same size, this paper proposes a new approach to main memory compression - Linearly Compressed Pages (LCP) - that avoids the performance degradation problem without requiring costly or energy-inefficient hardware. We show that any compression algorithm can be adapted to fit the requirements of LCP, and we specifically adapt two previously proposed compression algorithms to LCP: Frequent Pattern Compression and Base-Delta-Immediate compression. Evaluations using benchmarks from SPEC CPU2006 and five server benchmarks show that our approach can significantly increase the effective memory capacity (69% on average). In addition to the capacity gains, we evaluate the benefit of transferring consecutive compressed cache lines between the memory controller and main memory. Our new mechanism considerably reduces the memory bandwidth requirements of most of the evaluated benchmarks (34% on average), and improves overall performance (6.1%/13.9%/10.7% for single-/two-/four-core workloads on average) compared to a baseline system that does not employ main memory compression. LCP also decreases energy consumed by the main memory subsystem (9.5% on average over the best prior mechanism).

Tetrisched: Space-Time Scheduling for Heterogeneous Datacenters

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

Carnegie Mellon University Parallel Data Lab Technical Report CMU-

continued on page 21

continued from page 20

PDL-13-112, December, 2013.

Tetrisched is a new scheduler that explicitly considers both job-specific preferences and estimated job runtimes in its allocation of resources. Combined, this information allows tetrisched to provide higher overall value to complex application mixes consolidated on heterogeneous collections of machines. Job-specific preferences, provided by tenants in the form of composable utility functions, allow tetrisched to understand which resources are preferred, and by how much, over other acceptable options. Estimated job runtimes allow tetrisched to plan ahead in deciding whether to wait for a busy preferred resource to become free or to assign a less preferred resource. Tetrisched translates this information, which can be provided automatically by middleware (our wizard) that understands the right SLOs, runtime estimates, and budgets, into a MILP problem that it solves to maximize overall utility. Experiments with a variety of job type mixes, workload intensities, degrees of burstiness, preference strengths, and input inaccuracies show that tetrisched consistently provides significantly better schedules than alternative approaches.

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

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

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

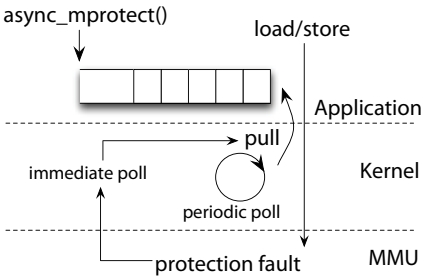
Distributed systems are complex to develop and administer, and performance problem diagnosis is particularly challenging. When performance degrades, the problem might be in any of the system's many components or could be a result of poor interactions among them. Recent research efforts have created tools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

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

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

TRIOS: Conference on Timely Results in Operating Systems. Held in conjunction with SOSP '13. Farmington, PA, November 3, 2013.

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



High-level operation of batched asynchronous memory protection.

Making Problem Diagnosis Work for Large-Scale, Production Storage Systems

Michael P. Kasick, Priya Narasimhan & Kevin Harms

Proceedings of the 27th Large Installation System Administration Conference (LISA '13), Washington, DC, November 2013.

Intrepid has a very-large, production GPFS storage system consisting of 128 file servers, 32 storage controllers, 1152 disk arrays, and 11,520 total disks. In such a large system, performance problems are both inevitable and difficult to troubleshoot. We present our experiences, of taking an automated problem diagnosis approach from proof-of-concept on a 12-server test-bench parallel-file-system cluster, and making it work on Intrepid's storage system. We also present a 15-month case study, of problems observed from the analysis of 624 GB of Intrepid's instrumentation data, in which we diagnose a variety of performance-related storage-system problems, in a matter of hours, as compared to the days or longer with manual approaches.

continued on page 22

continued from page 21

Measuring Password Guessability for an Entire University

Michelle L. Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay & Blase Ur

In CCS 2013: ACM Conference on Computer and Communications Security, November 2013.

Despite considerable research on passwords, empirical studies of password strength have been limited by lack of access to plaintext passwords, small data sets, and password sets specifically collected for a research study or from low-value accounts. Properties of passwords used for high-value accounts thus remain poorly understood.

We fill this gap by studying the single-sign-on passwords used by over 25,000 faculty, staff, and students at a research university with a complex password policy. Key aspects of our contributions rest on our (indirect) access to plaintext passwords. We describe our data collection methodology, particularly the many precautions we took to minimize risks to users. We then analyze how guessable the collected passwords would be during an offline attack by subjecting them to a state-of-the-art password cracking algorithm. We discover significant correlations between a number of demographic and behavioral factors and password strength. For example, we find that users associated with the computer science school make passwords more than 1.5 times as strong as those of users associated with the business school. In addition, we find that stronger passwords are correlated with a higher rate of errors entering them.

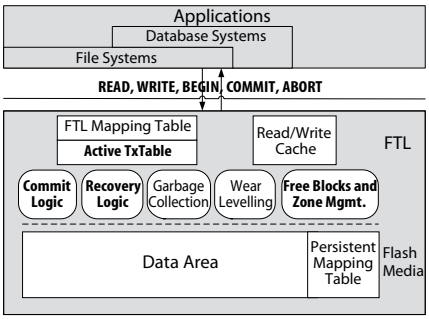
We also compare the guessability and other characteristics of the passwords we analyzed to sets previously collected in controlled experiments or leaked from low-value accounts. We find more consistent similarities between the university passwords and passwords collected for research studies under similar composition policies than we do between the university passwords and subsets of passwords leaked from low-value accounts that happen to comply with the same policies.

LightTx: A Lightweight Transactional Design in Flash-based SSDs to Support Flexible Transactions

Youyou Lu, Jiwu Shuy, Jia Guo, Shuai Li & Onur Mutlu

The 32nd IEEE International Conference on Computer Design (ICCD13). October 6-9, 2013, Ashville, NC, USA.

Flash memory has accelerated the architectural evolution of storage systems with its unique characteristics compared to magnetic disks. The no-overwrite property of flash memory has been leveraged to efficiently support transactions, a commonly used mechanism in systems to provide consistency. However, existing transaction designs embedded in flash-based Solid State Drives (SSDs) have limited support for transaction flexibility, i.e., support for different isolation levels between transactions, which is essential to enable different systems to make tradeoffs between performance and consistency. Since they provide support for only strict isolation between transactions, existing designs lead to a reduced number of on-the-fly requests and therefore cannot exploit the abundant internal parallelism of an SSD. There are two design challenges that need to be overcome to support flexible transactions: (1) enabling a transaction commit protocol that supports parallel execution of transactions; and (2) efficiently tracking the state of transactions that have pages scattered over different locations due to parallel allocation of pages.



The LightTx Architecture.

In this paper, we propose LightTx to address these two challenges. LightTx supports transaction flexibility using a lightweight embedded transaction design. The design of LightTx is based on two key techniques. First, LightTx uses a commit protocol that determines the transaction state solely inside each transaction (as opposed to having dependencies between transactions that complicate state tracking) in order to support parallel transaction execution. Second, LightTx periodically retires the dead transactions to reduce transaction state tracking cost. Experiments show that LightTx provides up to 20.6% performance improvement due to transaction flexibility. LightTx also achieves nearly the lowest overhead in garbage collection and mapping persistence compared to existing embedded transaction designs.

There Is More Consensus in Egalitarian Parliaments

Iulian Moraru, David G. Andersen & Michael Kaminsky

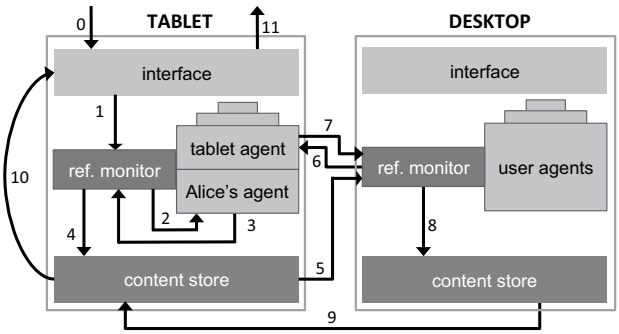
Proceedings of the 24th ACM Symposium on Operating Systems Principles (SOSP'13), November 3-6, 2013, Nemacolin Woodlands Resort, Farmington, PA.

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

continued on page 23

continued from page 22

achieving high throughput); and (3) graceful performance degradation when replicas are slow or crash. Egalitarian Paxos is to our knowledge the first protocol to simultaneously 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. We prove Egalitarian Paxos's properties theoretically and demonstrate its advantages empirically through an implementation running on Amazon EC2.



Access-control architecture. Using her tablet, Alice requests to open a file stored on the desktop (0). The interface component forwards this request to the reference monitor to be validated (1). The local monitor produces a challenge, which is proved by Alice's local agent (2-3), then asks the content store for the file (4). The content store requests the file from the desktop (5), triggering a challenge from the desktop's reference monitor (6). Once the tablet's agent proves that the tablet is authorized to receive the file (7), the desktop monitor instructs the desktop content store to send it to the tablet (8). The tablet's content store returns the file to Alice via the interface component (10-11).

Toward Strong, Usable Access Control for Shared Distributed Data

Michelle L. Mazurek, Yuan Liang, Manya Sleeper, Lujo Bauer, Gregory R. Ganger, Nitin Gupta & Michael K. Reiter

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-13-110. June 2013.

As non-expert users produce increasing amounts of personal digital data, providing them with usable access control becomes critical. Current approaches are frequently unsuccessful, either because they insufficiently protect data or confuse users about policy specification. We present a distributed file-system access-control infrastructure designed to match users' mental models while providing principled security. Our design combines semantic, tag-based policy specification with logic-based access control, enabling flexible support for intuitive policies while providing high assurance of correctness. We support private and conflicting tags, decentralized policy enforcement, and unforgeable audit records. Our logic can express a variety of policies that map well to real users' needs. To evaluate our design, we develop a set of detailed, realistic case studies drawn from prior research into users' access-control needs. The case studies can also be applied to other systems in the personal access-control space. Using simulated traces generated from the case studies, we demonstrate that our prototype implementation can enforce users' policies with acceptable overhead.

Challenges in Security and Privacy for Mobile Edge-Clouds

Jiaqi Tan, Rajeev Gandhi, Priya Narasimhan

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-13-113. October, 2013.

Mobile devices such as smartphones and tablets are ubiquitous today, and many of them possess significant computation power, powerful sensors such as high-resolution cameras and GPS sensors, and a wealth of sensor data such as photos, videos, and location information. Collections of mobile devices in close geographical proximity present both opportunities and challenges for mobile applications: the enormous collection of mobile devices presents an extremely rich source of user-generated content as well as collective computation power, but these devices are mutually distrusting, and security and privacy concerns, amongst many other obstacles, prevent users from cooperating with other distrusting entities to exploit both the available computation power and data. In this paper, we articulate and describe some of the security and privacy challenges which currently prevent us from leveraging the collective data and computational power available in collections of mobile devices belonging to mutually distrusting users. By addressing these security and privacy challenges, we envision that a new class of applications can be developed which leverage the collective mobile devices available in close geographical proximity: mutually distrusting users would be willing to participate in such public computations with sufficient security and privacy safeguards, enabling novel applications.

Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation

Yu Cai, Onur Mutlu, Erich F. Haratsch & Ken Mai

Proceedings of the 32nd IEEE International Conference on Computer Design (ICCD), Asheville, NC, October 2013.

As NAND flash memory continues to scale down to smaller process technology nodes, its reliability and endurance

continued on page 24

continued from page 23

are degrading. One important source of reduced reliability is the phenomenon of program interference: when a flash cell is programmed to a value, the programming operation affects the threshold voltage of not only that cell, but also the other cells surrounding it. This interference potentially causes a surrounding cell to move to a logical state (i.e., a threshold voltage range) that is different from its original state, leading to an error when the cell is read. Understanding, characterizing, and modeling of program interference, i.e., how much the threshold voltage of a cell shifts when another cell is programmed, can enable the design of mechanisms that can effectively and efficiently predict and/or tolerate such errors.

In this paper, we provide the first experimental characterization of and a realistic model for program interference in modern MLC NAND flash memory. To this end, we utilize the read-retry mechanism present in some state-of-the-art 2Y-nm (i.e., 20-24nm) flash chips to measure the changes in threshold voltage distributions of cells when a particular cell is programmed. Our results show that the amount of program interference received by a cell depends on 1) the location of the programmed cells, 2) the order in which cells are programmed, and 3) the data values of the cell that is being programmed as well as the cells surrounding it. Based on our experimental characterization, we develop a new model that predicts the amount of program interference as a function of threshold voltage values and changes in neighboring cells. We devise and evaluate one application of this model that adjusts the read reference voltage to the predicted threshold voltage distribution with the goal of minimizing erroneous reads. Our analysis shows that this new technique can reduce the raw flash bit error rate by 64% and thereby improve flash lifetime by 30%. We hope that the understanding and models developed in this paper lead to other error tolerance mechanisms for future flash memories.

Memory-Efficient GroupBy-Aggregate using Compressed Buffer Trees

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

SoCC'13, Oct. 01–03 2013, Santa Clara, CA, USA.

Memory is rapidly becoming a precious resource in many data processing environments. This paper introduces a new data structure called a Compressed Buffer Tree (CBT). Using a combination of buffering, compression, and lazy aggregation, CBTs can improve the memory efficiency of the GroupBy-Aggregate abstraction which forms the basis of many data processing models like MapReduce and databases. We evaluate CBTs in the context of MapReduce aggregation, and show that CBTs can provide significant advantages over existing hash-based aggregation techniques: up to 2X less memory and 1.5X the throughput, at the cost of 2.5X CPU.

Memory Scaling: A Systems Architecture Perspective

Onur Mutlu

MemCon 2013 (MEMCON), Santa Clara, CA, August 2013.

The memory system is a fundamental performance and energy bottleneck in almost all computing systems. Recent system design, application, and technology trends that require more capacity, bandwidth, efficiency, and predictability out of the memory system make it an even more important system bottleneck. At the same time, DRAM technology is experiencing difficult technology scaling challenges that make the maintenance and enhancement of its capacity, energy-efficiency, and reliability significantly more costly with conventional techniques. In this paper, after describing the demands and challenges faced by the memory system, we examine some promising research and design directions to overcome challenges posed by memory scaling. Specifically, we survey three key solution directions: 1) enabling new DRAM architectures, functions, interfaces, and better integration of the DRAM and the rest of the system, 2) designing a memory system that employs emerging memory technologies and takes advantage of multiple different technologies, 3) providing predictable performance and QoS to applications sharing the memory system. We also briefly describe our ongoing related work in combating scaling challenges of NAND flash memory.

I/O Acceleration with Pattern Detection

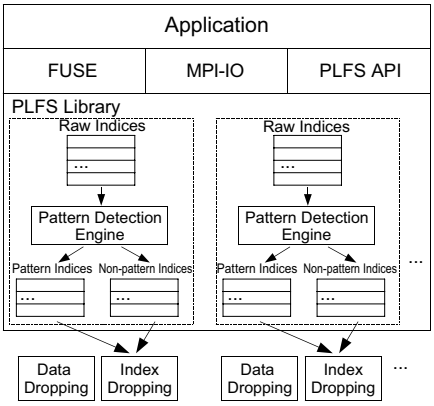
Jun He, John Bent, Aaron Torres, Gary Grider, Garth Gibson, Carlos Maltzahn & Xian-He Sun

The 22nd Int. ACM Symposium on High Performance Parallel and Distributed Computing (HPDC'13), New York City, June 17-21, 2013.

The I/O bottleneck in high-performance computing is becoming worse as application data continues to grow. In this work, we explore how patterns of I/O within these applications can significantly affect the effectiveness of the underlying storage systems and how these same patterns can be utilized to improve many aspects of the I/O stack and mitigate the I/O bottleneck. We offer three main contributions in this paper. First, we develop and evaluate algorithms by which I/O patterns can be efficiently discovered and described. Second, we implement one such algorithm to reduce the metadata quantity in a virtual parallel file system by up to several orders of magnitude, thereby increasing the performance of writes and reads by up to 40 and 480 percent respectively. Third, we build a prototype file system with pattern-aware prefetching and evaluate it to show a 46 percent reduction

continued on page 25

continued from page 24



Pattern PLFS index framework, when writing, Pattern PLFS buffers traditional indices in raw index buffers for each process. After the buffer is full or at the time of closing, a pattern discovering engine starts processing the raw indices and puts the generated pattern structure entries to pattern index buffer and non-pattern ones to non-pattern indices. At the end, the entries will be written to pattern index files.

in I/O latency. Finally, we believe that efficient pattern discovery and description, coupled with the observed predictability of complex patterns within many high-performance applications, offers significant potential to enable many additional I/O optimizations.

PRObE: A Thousand-Node Experimental Cluster for Computer Systems Research

Garth Gibson, Gary Grider, Andree Jacobson & Wyatt Lloyd

USENIX ;login:, v 38, n 3, June 2013.

If you have ever aspired to create a software system that can harness a thousand computers and perform some impressive feat, you know the dismal prospects of finding such a cluster ready and waiting for you to make magic with it. Today, however, if you are a systems researcher and your promised feat is impressive enough, there is such a resource available online: PRObE. This article is an introduction to and call for proposals for use of the PRObE facilities.

Hadoop's Adolescence: An Analysis of Hadoop Usage in Scientific Workloads

Kai Ren, YongChul Kwon, Magdalena Balazinska & Bill Howe

Very Large Data Bases (VLDB), August, 2013.

We analyze Hadoop workloads from three different research clusters from a user-centric perspective. The goal is to better understand data scientists' use of the system and how well the use of the system matches its design. Our analysis suggests that Hadoop usage is still in its adolescence. We see underuse of Hadoop features, extensions, and tools. We see significant diversity in resource usage and application styles, including some interactive and iterative workloads, motivating new tools in the ecosystem. We also observe significant opportunities for optimizations of these workloads. We find that job customization and configuration are used in a narrow scope, suggesting the future pursuit of automatic tuning systems. Overall, we present the first user-centered measurement study of Hadoop and find significant opportunities for improving its efficient use for data scientists.

A Proof of Correctness for Egalitarian Paxos

Iulian Moraru, David G. Andersen & Michael Kaminsky

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-13-111. August 2013.

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.

Specialized Storage for Big Numeric Time Series

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

Proceedings of the 5th Workshop on Hot Topics in Storage and File Systems, June 2013.

Numeric time series data has unique storage requirements and access patterns that can benefit from specialized support, given its importance in Big Data analyses. Popular frameworks and databases focus on addressing other needs, making them a suboptimal fit. This paper describes the support needed for numeric time series, suggests an architecture for efficient time series storage, and illustrates its potential for satisfying key requirements.

An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms

Jamie Liu, Ben Jaiyen, Yoongu Kim, Chris Wilkerson & Onur Mutlu

Proceedings of the 40th International Symposium on Computer Architecture (ISCA), Tel-Aviv, Israel, June 2013.

DRAM cells store data in the form of charge on a capacitor. This charge leaks off over time, eventually causing data

continued on page 26

continued from page 25

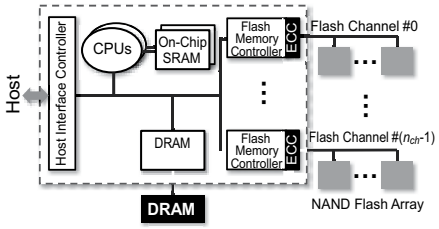
to be lost. To prevent this data loss from occurring, DRAM cells must be periodically refreshed. Unfortunately, DRAM refresh operations waste energy and also degrade system performance by interfering with memory requests. These problems are expected to worsen as DRAM density increases.

The amount of time that a DRAM cell can safely retain data without being refreshed is called the cell's retention time. In current systems, all DRAM cells are refreshed at the rate required to guarantee the integrity of the cell with the shortest retention time, resulting in unnecessary refreshes for cells with longer retention times. Prior work has proposed to reduce unnecessary refreshes by exploiting differences in retention time among DRAM cells; however, such mechanisms require knowledge of each cell's retention time. In this paper, we present a comprehensive quantitative study of retention behavior in modern DRAMs. Using a temperature-controlled FPGA-based testing platform, we collect retention time information from 248 commodity DDR3 DRAM chips from five major DRAM vendors. We observe two significant phenomena: data pattern dependence, where the retention time of each DRAM cell is significantly affected by the data stored in other DRAM cells, and variable retention time, where the retention time of some DRAM cells changes unpredictably over time. We discuss possible physical explanations for these phenomena, how their magnitude may be affected by DRAM technology scaling, and their ramifications for DRAM retention time profiling mechanisms.

Active Disk Meets Flash: A Case for Intelligent SSDs

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

Proceedings of the ACM Int'l Conference on Supercomputing (ICS), Eugene, OR, June 2013.



Time frame	Characteristics
2007–2008	4-way, 4 channels, 30–80 MB/s R/W performance; mostly SLC flash based;
2008–2009	8–10 channels, 150–200+ MB/s performance (SATA, consumer); 16+ channels, 600+ MB/s performance (PCI-e, enterprise); use of MLC flash in consumer products;
2009–2010	16+ channels, 200–300+ MB/s performance (SATA 6 Gbps); 20+ channels, 1+ GB/s performance (PCI-e); adoption of MLC in enterprise products;
2010–	16+ channels; wider acceptance of PCI-e;

General architecture of an SSD (top): The dashed box is the boundary of the controller chip. SSD evolution with new host interface standards (bottom).

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

TABLEFS: Enhancing Metadata Efficiency in the Local File System

Kai Ren & Garth Gibson

2013 USENIX Annual Technical Conference, June 26-28, 2013, San Jose, CA.

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.

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 underlying 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 27

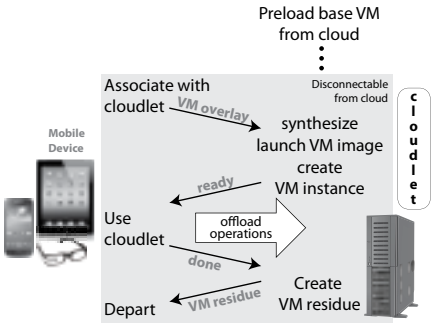
continued from page 26

Just-in-Time Provisioning for Cyber Foraging

Kiryong Ha, Padmanabhan Pillai, Wolfgang Richter, Yoshihisa Abe & Mahadev Satyanarayanan

MobiSys'13, June 25–28, 2013, Taipei.

Cloud offload is an important technique in mobile computing. VMbased cloudlets have been proposed as offload sites for the resource-intensive and latency-sensitive computations typically associated with mobile multimedia applications. Since cloud offload relies on precisely-configured back-end software, it is difficult to support at global scale across cloudlets in multiple domains. To address this problem, we describe just-in-time (JIT) provisioning of cloudlets under the control of an associated mobile device. Using a suite of five representative mobile applications, we demonstrate a prototype system that is capable of provisioning a cloudlet with a non-trivial VM image in 10 seconds. This speed is achieved through dynamic VM synthesis and a series of optimizations to aggressively reduce transfer costs and startup latency.



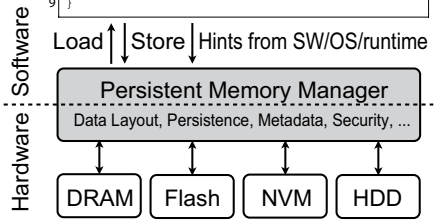
Dynamic VM Synthesis from Mobile Device.

A Case for Efficient Hardware/Software Cooperative Management of Storage and Memory

Justin Meza, Yixin Luo, Samira Khan, Jishen Zhao, Yuan Xie & Onur Mutlu

5th Workshop on Energy-Efficient Design (WEED), Tel-Aviv, June 2013.

```
1 int main(void){
2   //data in file.dat is persistent
3   FILE myData = "file.dat";
4   myData = new int[64];
5 }
6 void updateValue(int n, int value){
7   FILE myData = "file.dat";
8   myData[n] = value; //value is persistent
9 }
```



Hardware-based storage and memory management. In the code listing above, the main function creates a new persistent object with the handle "file.dat" and sets its contents to a newly-allocated array of 64 integers. Later, in the `updateValue` function, the persistent object with the same handle ("file.dat") is accessed, and the value of one of its elements is updated, just as it would be if it were allocated in memory, except that now the PMM ensures that its data is mapped to persistent memory, can be located efficiently, and will remain persistent even if the machine were to crash after the program updates the value.

Most applications manipulate persistent data, yet traditional systems decouple data manipulation from persistence in a two-level storage model. Programming languages and system software manipulate data in one set of formats in volatile main memory (DRAM) using a load/store interface, while storage systems maintain persistence in another set of formats in non-volatile memories, such as Flash and hard disk drives in traditional systems, using a file system interface. Unfortunately, such an approach suffers from the system performance and energy overheads of locating data, moving data, and translating data between the different formats of these two levels of storage that are accessed via two vastly different interfaces. Yet today, new non-volatile memory (NVM) technologies show the promise of storage capacity and endurance similar to or better than Flash at latencies comparable to DRAM, making them prime candidates for providing applications a persistent single-level store with a single load/store interface to access all system data. Our key insight is that in future systems equipped with NVM, the energy consumed executing operating system and file system code to access persistent data in traditional systems becomes an increasingly large contributor to total energy. The goal of this work is to explore the design of a Persistent Memory Manager that coordinates the management of memory and storage under a single hardware unit in a single address space. Our initial simulation-based exploration shows that such a system with a persistent memory can improve energy efficiency and performance by eliminating the instructions and data movement traditionally used to perform I/O operations.

Building a High-Performance Metadata Service by Reusing Scalable I/O Bandwidth

Kai Ren, Swapnil Patil, Kartik Kulkarni, Adit Madan & Garth Gibson

Carnegie Mellon University Parallel Data Laboratory Technical Report CMU-PDL-13-107, May 2013.

Modern parallel and cluster file systems provide highly scalable I/O bandwidth by enabling highly parallel access to file data. Unfortunately metadata access does not benefit from parallel data transfer, so metadata performance scaling is less common. To support metadata-intensive workloads, we offer a middleware design that layers on top of existing cluster file systems, adds support for load balanced and high-performance metadata operations without sacrificing data bandwidth. The core idea is to integrate a distributed indexing mechanism with a metadata optimized on-disk Log-Structured Merge tree layout. The integration requires several optimizations including cross-server

continued on page 28

continued from page 27

split operations with minimum data migration, and decoupling of data and metadata paths. To demonstrate the feasibility of our approach, we implemented a prototype middleware layer GIGA+TableFS and evaluated it with a Panasas parallel file system. GIGA+TableFS improves metadata performance of PanFS by as much an order of magnitude, while still performing comparably on data-intensive workloads.

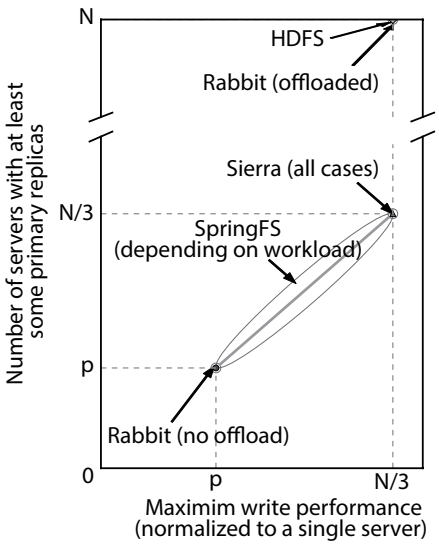
SpringFS: Bridging Agility and Performance in Elastic Distributed Storage

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

12th USENIX Conference on File and Storage Technologies (FAST '14), Santa Clara, CA, February 17–20, 2014.

Elastic storage systems can be expanded or contracted to meet current demand, allowing servers to be turned off or used for other tasks. However, the usefulness of an elastic distributed storage system is limited by its agility: how quickly it can increase or decrease its number of servers. Due to the large amount of data they must migrate during elastic resizing, state-of-the-art designs usually have to make painful tradeoffs among performance, elasticity and agility.

This paper describes an elastic storage system, called SpringFS, that can quickly change its number of active servers, while retaining elasticity and performance goals. SpringFS uses a novel technique, termed bounded write offloading, that restricts the set of servers where writes to overloaded servers are redirected. This technique, combined with the read offloading and passive migration policies used in SpringFS, minimizes the work needed before deactivation or activation of servers. Analysis of real-world traces from Hadoop deployments at Facebook and various Cloudera customers and experiments with the SpringFS prototype confirm SpringFS's agility, show that it reduces the amount of data migrated for elastic resizing by up to two orders of magnitude, and show that it cuts the percentage of active servers required by 67–82%, outdoing state-of-the-art designs by 6–120%.



Elastic storage system comparison in terms of agility and performance. *N* is the total size of the cluster. *p* is the number of primary servers in the equal-work data layout. Servers with at least some primary replicas cannot be deactivated without first moving those primary replicas. SpringFS provides a continuum between Sierra's and Rabbit's (when no offload) single points in this tradeoff space. When Rabbit requires offload, SpringFS is superior at all points. Note that the y-axis is discontinuous.

2013 PDL Workshop and Retreat.