FALL UPDATE
PDL PACKET

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • FALL 2015
http://www.pdl.cmu.edu/

PDL CONSORTIUM MEMBERS

Actifio
Avago Technologies
EMC
Facebook
Google
Hewlett-Packard Labs
Hitachi
Intel
Microsoft Research
MongoDB
NetApp
Oracle Corporation
Samsung Information Systems America
Seagate Technology
Symantec Corporation
Western Digital

CONTENTS

THE PDL PACKET

EDITOR
Joan Digney
CONTACTS
Greg Ganger
PDL Director
Bill Courtright
PDL Executive Director
Karen Lindenfelser
PDL Administrative Manager
The Parallel Data Laboratory
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3891
TEL 412-268-6716
FAX 412-268-3010

http://www.pdl.cmu.edu/Publications/

SELECTED RECENT PUBLICATIONS

Managed Communication and Consistency for Fast Data-Parallel Iterative Analytics

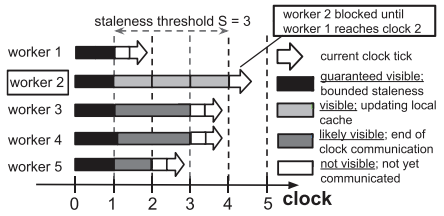
Jinliang Wei, Wei Dai, Aurick Qiao, Qirong Ho, Henggang Cui, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson & Eric P. Xing

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

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

This paper presents Bösen, a system that maximizes the network communication efficiency under a given intermachine network bandwidth budget to minimize accumulated error, while ensuring theoretical convergence guarantees for large-scale data-parallel ML applications. Furthermore, Bösen prioritizes messages that are most significant to algorithm convergence, further enhancing algorithm convergence. Finally, Bösen is the first distributed implementation of the recently presented adaptive revision algorithm, which provides orders of magnitude improvement over a carefully tuned

continued on page 6



An execution of 5 workers under bounded staleness (without communication management). The system consists of 5 workers, with staleness threshold S = 3; "iteration t" refers to iteration starting from t. Worker 2 is currently running in iteration 4 and thus according to bounded staleness, it is guaranteed to observe all updates generated before (exclusively) iteration 4−3 = 1 (black). It may also observe local updates (lt. gray) as updates can be optionally applied to local parameter cache. Updates that are generated in completed iterations (i.e. clock ticks) by other workers (dk. gray) are highly likely visible as they are propagated at the end of each clock. Updates generated in incomplete iterations (white) are not visible as they are not yet communicated. Such updates could be made visible under managed communication depending on the bandwidth budget.

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

Wolfgang Richter

Carnegie Mellon University, SCS

Ph.D. Dissertation
September 18, 2015

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

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

Andy Pavlo, (CMU) and Jim Hunt (Facebook) talk database technology during a PDL Visit Day poster session.

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

Hyeontaek Lim

Carnegie Mellon University, SCS

Ph.D. Dissertation
July 20, 2015

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

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

DISSERTATION ABSTRACT:
Providing High and Predictable Performance in Multicore Systems through Shared Resource Management

Lavanya Subramanian

Carnegie Mellon University, SCS

Ph.D. Dissertation
April 28, 2015

Multiple applications executing concurrently on a multicore system interfere with each other at the different

continued on page 3

continued from page 2

shared resources such as main memory and shared caches. Such inter-application interference, if uncontrolled, results in high system performance degradation and unpredictable application slowdowns. While previous work has proposed application-aware memory scheduling as a solution to mitigate inter-application interference and improve system performance, previously proposed memory scheduling techniques incur high hardware complexity and unfairly slowdown some applications. Furthermore, previously proposed memory-interference mitigation techniques are not designed to provide predictable application performance.

This dissertation seeks to achieve high and predictable performance in multicore systems by mitigating and quantifying the impact of shared resource interference. First, towards mitigating memory interference and achieving high performance, we propose the Blacklisting memory scheduler. We observe that ranking applications individually with a total order based on memory access characteristics, like previous schedulers do, leads to high hardware cost, while also causing unfair application slowdowns. The Blacklisting memory scheduler overcomes these shortcomings based on two key observations. First, we observe that, to mitigate interference, it is sufficient to separate applications into only two groups, one containing applications that are vulnerable to interference and another containing applications that cause interference, instead of ranking individual applications with a total order. Vulnerable-to-interference group is prioritized over the interference-causing group. Second, we show that this grouping can be efficiently performed by simply counting the number of consecutive requests served from each application -- an application

Dana Van Aken and Xiaolin Zang (CMU) discuss "BenchPress: Dynamic Workload Control in the OLTP-Bench Testbed" at the PDL Spring Visit Day.

that has a large number of consecutive requests served is dynamically classified as interference-causing. The Blacklisting memory scheduler, designed based on these insights, achieves high system performance and fairness, while incurring significantly lower complexity than state-of-the-art application-aware schedulers.

Next, towards quantifying the impact of memory interference and achieving predictable performance in the presence of memory bandwidth interference, we propose the Memory Interference induced Slowdown Estimation (MISE) model. The MISE model estimates application slowdowns due to memory interference based on two observations. First, the performance of a memory-bound application is roughly proportional to the rate at which its memory requests are served, suggesting that request-service-rate can be used as a proxy for performance. Second, when an application's requests are prioritized over all other applications' requests, the application experiences very little interference from other applications. This provides a means for predicting the uninterfered request-service-rate of an application while it is run alongside other applications. Using the above observations, MISE predicts the slowdown of an application as the ratio of its uninterfered and interfered request service rates. We propose simple changes to the above model to predict the slowdown of non-memory-bound applications. We propose and demonstrate two use cases that can leverage MISE to achieve predictable performance and high overall performance/fairness.

Finally, we seek to quantify the impact of shared cache interference on application slowdowns, in addition to memory bandwidth interference. Towards this end, we propose the Application Slowdown Model (ASM). ASM builds on MISE and observes that the performance of an application is strongly correlated with the rate at which the application accesses the shared cache. This is a more general observation than that of MISE and holds for all applications, thereby enabling the estimation of slowdown for any application as the ratio of the uninterfered to the interfered shared cache access rate. This reduces the problem of estimating slowdown to estimating the shared cache access rate of the application had it been run alone on the system. ASM periodically estimates each application's cache-access-rate-alone by minimizing interference at the main memory and quantifying interference at the shared cache. We propose and demonstrate several use cases of ASM that leverage it to provide predictable performance and improve performance and fairness.

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

Junchen Jiang, SCS

July 2, 2015

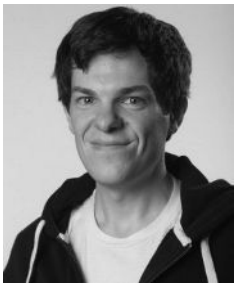
Transport layer and application layer of network stack use end-to-end

continued on page 12

September 2015
Samira Khan now Faculty at University of Virginia

Best wishes to Samira as she joins the CS department at the University of Virginia as an Assistant Professor. Samira is mainly interested in Computer Architecture and Computer systems, especially in building new systems by rethinking the traditional assumptions in abstraction and separation of responsibilities in different system layers and redesigning interfaces with new interaction and collaboration to solve systems/architecture research problems.

September 2015
Three PDL Faculty Receive Google Faculty Research Awards



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

Latency SLOs" (Google), and Lorrie Cranor will focus her award on research in the Human-Computer Interaction area.

September 2015
Two PDL Faculty Receive Facebook Faculty Awards

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

September 2015
New PDL Faculty!

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

September 2015
Best Paper Award at MobiArch!

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

September 2015
Jiaqi Tan Wins FMCAD Award!

Congratulations Jiaqi Tan, for winning the Best Contribution Award at the Student

continued on page 5

continued from page 4

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

August 2015

Best Paper Award at SoCC!

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

May 2015

Alexey Tumanov Receives ECE's Graduate Student Teaching Assistant Award

Congratulations to Alexey for receiving ECE's Outstanding Graduate Student Teaching Assistant Award for his efforts on 15-719: Advanced Cloud Computing, taught by Garth Gibson and Majd Sakr during the fall semester of 2014. In their letter of nomination, Professors Sakr and Gibson cited Alexey's hard work, innovation, and commitment to student success,

Alexey receives his Outstanding Teaching Assistant Award from Professor Marculescu at the 2015 ECE Spring Commencement ceremonies.

describing it as "unparalleled". Alexey "went way beyond the call of duty, supported the students with a pleasant constructive engagement style and built a project [that] will certainly [be] reused next year."

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

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

May 2015

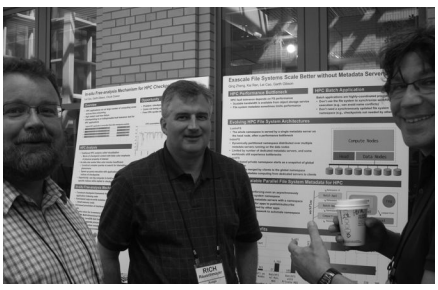
NVIDIA Graduate Fellowship Winner

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

May 2015

Congratulations Wolf and Debjani!

Best wishes for a long, happy life together for Wolf Richter and Debjani Biswas. They were married on May 20th in Kolkata, India and now live in Boston, MA, where Wolf works on a startup.



From L to R, Bruce Wilson, Rich Rauschmayer, and Horia Simionescu, all of Avago Technologies, enjoying a poster session at the 2015 PDL Spring Visit Day.

continued from page 1

fixed schedule of step size refinements. Experiments on two clusters with up to 1024 cores show that our mechanism significantly improves upon static communication schedules.

Exploiting Inter-Warp Heterogeneity to Improve GPGPU Performance

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

Proceedings of the The 24th International Conference on Parallel Architectures and Compilation Techniques (PACT 2015), San Francisco, October 2015. In a GPU, all threads within a warp execute the same instruction in lockstep. For a memory instruction, this can lead to memory divergence: the memory requests for some threads are serviced early, while the remaining requests incur long latencies. This divergence stalls the warp, as it cannot execute the next instruction until all requests from the current instruction complete.

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

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

Krowd: A Key-Value Store for Crowded Venues

Utsav Drolia, Nathan Mickulicz, Rajeev Gandhi & Priya Narasimhan

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

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

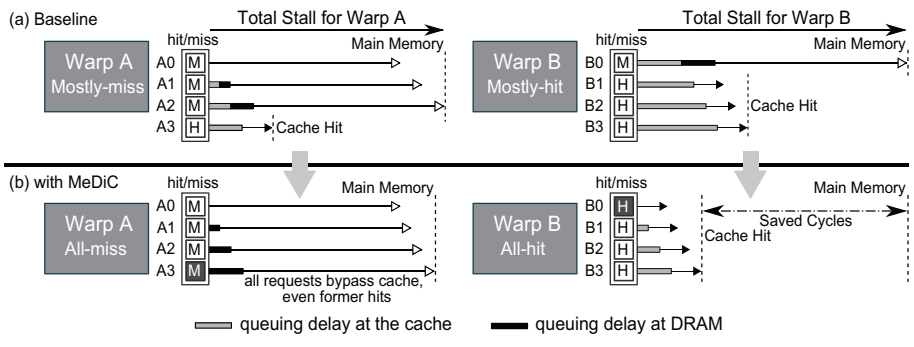
ShardFS vs. IndexFS: Replication vs. Caching Strategies for Distributed Metadata Management in Cloud Storage Systems

Lin Xiao, Kai Ren, Qing Zheng & Garth Gibson

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

The rapid growth of cloud storage systems calls for fast and scalable namespace processing. While few commercial file systems offer anything better than federating individually non-scalable namespace servers, a recent academic file system, IndexFS, demonstrates scalable namespace processing based on client caching of directory entries and permissions (directory lookup state) with no per-client state in servers. In this paper we explore explicit replication of directory lookup

continued on page 7



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

continued from page 6

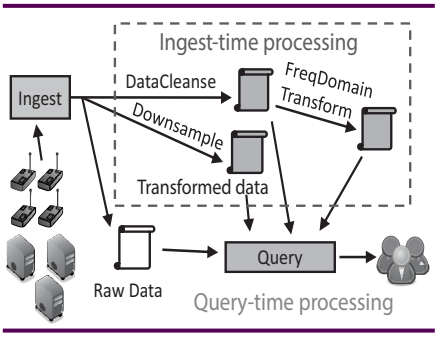
state in all servers as an alternative to caching this information in all clients. Both eliminate most repeated RPCs to different servers in order to resolve hierarchical permission tests. Our realization for server replicated directory lookup state, ShardFS, employs a novel file system specific hybrid optimistic and pessimistic concurrency control favoring single object transactions over distributed transactions. Our experimentation suggests that if directory lookup state mutation is a fixed fraction of operations (strong scaling for metadata), server replication does not scale as well as client caching, but if directory lookup state mutation is proportional to the number of jobs, not the number of processes per job, (weak scaling for metadata), then server replication can scale more linearly than client caching and provide lower 70 percentile response times as well.

Using Data Transformations for Low-latency Time Series Analysis

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

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

Time series analysis is commonly used when monitoring data centers, networks, weather, and even human patients. In most cases, the raw time series data is massive, from millions to billions of data points, and yet interactive analyses require low (e.g., subsecond) latency. Aperture transforms raw time series data, during ingest, into compact summarized representations that it can use to efficiently answer queries at runtime. Aperture handles a range of complex queries, from correlating hundreds of lengthy time series to predicting anomalies in the data. Aperture achieves much of its high performance by executing queries on data summaries, while providing a bound on the information lost when transforming data. By doing so,



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

Aperture can reduce query latency as well as the data that needs to be stored and analyzed to answer a query. Our experiments on real data show that Aperture can provide one to four orders of magnitude lower query response time, while incurring only 10% ingest time overhead and less than 20% error in accuracy.

Reducing Replication Bandwidth for Distributed Document Databases

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

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

With the rise of large-scale, Web-based applications, users are increasingly adopting a new class of document-oriented database management systems (DBMSs) that allow for rapid prototyping while also achieving scalable performance. Like for other distributed storage systems, replication is important for document DBMSs in order to guarantee availability. The bandwidth required to keep replicas synchronized is expensive and often becomes a bottleneck. As such, there is a strong need to reduce the replication bandwidth, especially for geo-replication scenarios where wide-area network (WAN) bandwidth is limited. This paper presents a deduplication system called sDedup that reduces the amount of data transferred over the network for replicated document DBMSs. sDedup uses similarity-based deduplication to remove redundancy in replication data by delta encoding against similar documents selected from the entire database. It exploits key characteristics of document-oriented workloads, including small item sizes, temporal locality, and the incremental nature of document edits. Our experimental evaluation of sDedup with three real-world datasets shows that it is able to achieve up to 38× reduction in data sent over the network, significantly outperforming traditional chunk-based deduplication techniques while incurring negligible performance overhead.

Scaling Up Clustered Network Appliances with ScaleBricks

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

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

This paper presents ScaleBricks, a new design for building scalable, clustered network appliances that must "pin" flow state to a specific handling node without being able to choose which node that should be. ScaleBricks applies a new, compact lookup structure to route packets directly to the appropriate handling node, without incurring the cost of multiple hops across the internal interconnect. Its lookup structure is many times smaller than the alternative approach of fully replicating a forwarding table onto all nodes. As a result, ScaleBricks is able to improve throughput and latency while simultaneously increasing the total number of flows that can be handled by such a cluster. This architecture is effective in practice: Used

continued on page 8

continued from page 7

to optimize packet forwarding in an existing commercial LTE-to-Internet gateway, it increases the throughput of a four-node cluster by 23%, reduces latency by up to 10%, saves memory, and stores up to 5.7x more entries in the forwarding table.

AUSPICE: Automated Safety Property Verification for Unmodified Executables

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

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

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

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

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

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

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

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

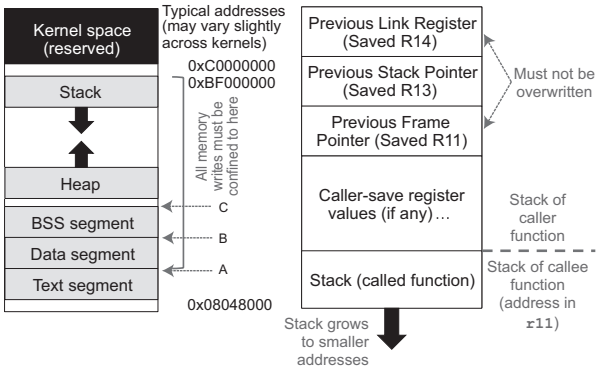
Caveat-Scriptor: Write Anywhere Shingled Disks

Saurabh Kadekodi, Swapnil Pimpale & Garth Gibson

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

The increasing ubiquity of NAND flash storage is forcing magnetic disks to accelerate the rate at which they lower price per stored bit. Magnetic recording technologists have begun to pack tracks so closely that writing one track cannot avoid disturbing the information stored in adjacent tracks [13]. Specifically, the downstream track will be at least partially over-

continued on page 9



Safety properties.

continued from page 8

written, or shingled by each write, as shown in Figure 1, and the upstream track will tolerate only a limited number of adjacent writes. Some data that was stored in the downstream track will be lost, forcing firmware or software to ensure that there was no data in those locations that might be read in the future.

Reducing Latency via Redundant Requests: Exact Analysis

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

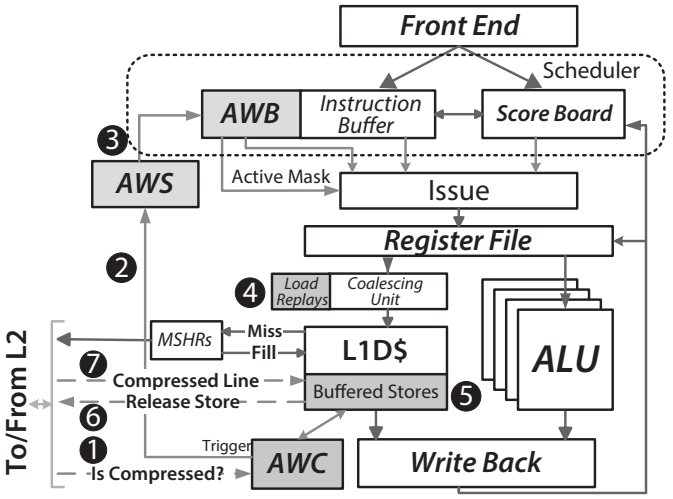
Proceedings of ACM Sigmetrics/Performance 2015 Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 15), Portland, OR. June 2015.

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

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

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

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



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

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

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

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

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

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

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

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

continued on page 10

continued from page 9

memory-bandwidth-sensitive GPGPU applications.

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

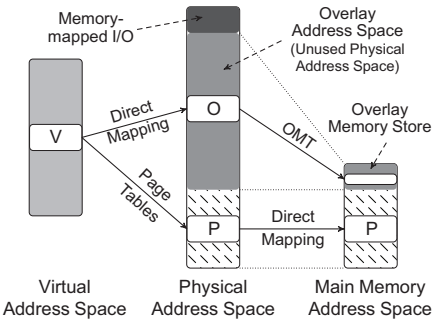
Vivek Seshadri, Gennady Pekhimenko, Olatunji Ruwase, Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, Todd C. Mowry & Trishul Chilimbi

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

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

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

We show that our framework can enable simple and efficient implementations of seven memory management techniques, each of which has a wide



Overview of our design. "Direct mapping" indicates that the corresponding mapping is implicit in the source address. OMT = Overlay Mapping Table

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

Optimal Scheduling for Jobs with Progressive Deadlines

Kristen Gardner, Sem Borst & Mor Harchol-Balter

IEEE INFOCOM 15, Hong Kong, April, 2015.

This paper considers the problem of server-side scheduling for jobs composed of multiple pieces with consecutive (progressive) deadlines. One example is server-side scheduling for video service, where clients request flows of content from a server with limited capacity, and any content not delivered by its deadline is lost. We consider the simultaneous goals of 1) minimizing overall loss, and 2) differentiating loss fractions across classes of flows in proportion to relative weights. State-of-the-art policies, like Discriminatory Processor Sharing and Weighted Fair Queueing, use a fixed static proportional allocation of service rate and fail to achieve both goals. The well-known Earliest Deadline First policy minimizes overall loss, but fails to provide proportional loss across flows, because it treats packets as independent jobs.

This paper introduces the Earliest Progressive Deadline First (EPDF) class of policies. We prove that all policies in this broad class minimize overall loss. Furthermore, we demonstrate that many EPDF policies accurately differentiate loss fractions in proportion to class weights, satisfying the second goal.

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

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

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

Trending search topics cause unpredictable query load spikes that hurt the end-user search experience, particularly the mobile one, by introducing longer delays. To understand how trending search topics are formed and evolve over time, we analyze 21 million queries submitted during periods where popular events caused search query volume spikes. Based on our findings, we design and evaluate PocketTrend, a system that automatically detects trending topics in real time, identifies the search content associated to the topics, and then intelligently pushes this content to users in a timely manner. In that way, PocketTrend enables a client-side search engine that can instantly answer user queries related to trending events, while at the same time reducing the impact of these

continued on page 11

continued from page 10

trends on the datacenter workload. Our results, using real mobile search logs, show that in the presence of a trending event, up to 13–17% of the overall search traffic can be eliminated from the datacenter, with as many as 19% of all users benefiting from PocketTrend.

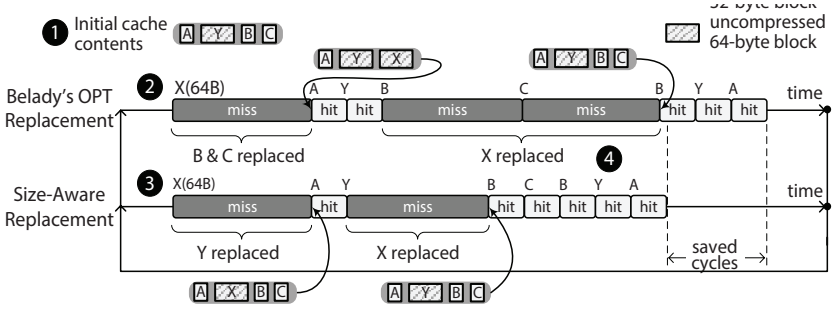
Exploiting Compressed Block Size as an Indicator of Future Reuse

Gennady Pekhimenko, Tyler Huberty, Rui Cai, Onur Mutlu, Phillip P. Gibbons, Michael A. Kozuch & Todd C. Mowry

Proc. of the 21st Int'l Symposium on High-Performance Computer Architecture (HPCA), Bay Area, CA, Feb. 2015.

We introduce a set of new Compression-Aware Management Policies (CAMP) for on-chip caches that employ data compression. Our management policies are based on two key ideas. First, we show that it is possible to build a more efficient management policy for compressed caches if the compressed block size is directly used in calculating the value (importance) of a block to the cache. This leads to Minimal-Value Eviction (MVE), a policy that evicts the cache blocks with the least value, based on both the size and the expected future reuse. Second, we show that, in some cases, compressed block size can be used as an efficient indicator of the future reuse of a cache block. We use this idea to build a new insertion policy called Size-based Insertion Policy (SIP) that dynamically prioritizes cache blocks using their compressed size as an indicator.

We compare CAMP (and its global variant G-CAMP) to prior on-chip cache management policies (both size-oblivious and size-aware) and find that our mechanisms are more effective in using compressed block size as an extra dimension in cache management decisions. Our results show that the proposed management policies (i) decrease off-chip bandwidth consumption (by 8.7% in single-core), (ii) decrease memory subsystem energy consumption (by 7.2% in single-core) for memory intensive workloads compared to the best prior mechanism, and (iii) improve performance (by 4.9%/9.0%/10.2% on average in single- /two-/four-core workload evaluations and up to 20.1%) CAMP is effective for a variety of compression algorithms and different cache designs with local and global replacement strategies.



Example demonstrating downside of not including block size information in replacement decisions. Initially (1), the cache contains three compressed (A, B, C) and one uncompressed (Y) block. Memory requests are made: X, A, Y, B, C, B, Y, A (2). After a request for X, Belady's algorithm (based on locality) evicts blocks B and C (creates 64 bytes of free space) for access in the future. Over the next four accesses, this results in two misses (B, C) and two hits (A, Y). A size-aware replacement policy makes the decision to retain B and C and evicts Y to make space for X (3). As a result, the cache experiences three hits (A, B, and C) and only one miss (Y) and hence outperforms Belady's optimal algorithm. Later (4), when there are three requests to blocks B, Y, and A , the final cache state becomes the same as the initial one. Hence, this example can represent steady state within a loop.

Toggle-Aware Compression for GPUs

Gennady Pekhimenko, Evgeny Bolotin, Mike O'Connor, Onur Mutlu, Todd C. Mowry & Stephen W. Keckler

IEEE Computer Architecture Letters (CAL), 2015.

Memory bandwidth compression can be an effective way to achieve higher system performance and energy efficiency in modern data-intensive applications by exploiting redundancy in data. Prior works studied various data compression techniques to improve both capacity (e.g., of caches and main memory) and bandwidth utilization (e.g., of the on-chip and off-chip interconnects). These works addressed two common shortcomings of compression: (i) compression/decompression overhead in terms of latency, energy, and area, and (ii) hardware complexity to support variable data size. In this paper, we make the new observation that there is another important problem related to data compression in the context of the communication energy efficiency: transferring compressed data leads to a substantial increase in the number of bit toggles (communication channel switchings from 0 to 1 or from 1 to 0). This, in turn, increases the dynamic energy consumed by on-chip and off-chip buses due to more frequent charging and discharging of the wires. Our results, for example, show that the bit toggle count increases by an average of 2.2X with some compression algorithms across 54 mobile GPU applications. We characterize and demonstrate this new problem across a wide variety of 221 GPU applications and six different compression algorithms. To mitigate the problem, we propose two new toggle-aware compression techniques: Energy Control and Metadata Consolidation. These techniques greatly reduce the bit toggle count impact of the six data compression algorithms we examine, while keeping most of their bandwidth reduction benefits.

continued from page 3

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

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

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

Suagata Ghose (PDL PostDoc), and Joy Arulraj (Grad Student) discuss their research.

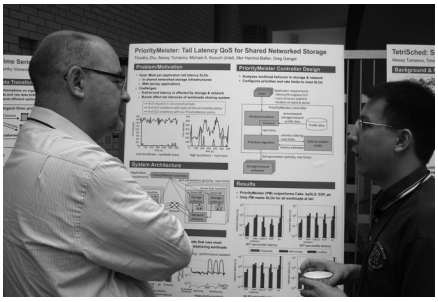
This thesis will present algorithms and system designs of centralized predictive control for both transport layer and application layer. We show that our approach can lead to better performance for TCP, Internet video and real-time communication applications like Skype. Our preliminary experiments have shown significant improvement of Internet video quality by centralized predictive control.

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

Alexey Tumanov, ECE

May 2015

Heterogeneity in datacenter hardware, software, and user objectives calls for new scheduling schemes to capture, aggregate, and leverage this information. Our proposed scheduler, TetriSched, explicitly considers cluster job-specific preferences in terms of where (space), when(time), and how (space-time shape) these jobs are scheduled. Spatial and temporal preferences combined allow TetriSched to provide higher overall value to complex data analytics mixes consolidated on heterogeneous collections of resources. First, we propose a principal building block—a new language called Space-Time Request Language (STRL). It enables the expression of these preferences in a general, extensible way by using a declarative, composable, algebraic structure with combinatorial primitives and allows TetriSched to understand which resources are preferred



Timmy Zhu describes "PriorityMeister: Tail Latency QoS for Shared Networked Storage" to Kent Foster (Facebook) at a Visit Day poster session.

and by how much, over other acceptable options. Estimated job runtimes for recurrent or profiled jobs allow TetriSched to consider deferred placement if the benefit of waiting for unavailable preferred resources exceeds the cost. Second, building on the generality of STRL, we propose an equally general STRL Compiler that automatically compiles STRL expressions into Mixed Integer Linear Programming (MILP) problems that can be aggregated and solved to maximize the overall value of shared cluster resources. Third, we propose a set of features that extend the scope and the practicality of TetriSched's deployment by analyzing and improving on its scalability, enabling and studying the efficacy of preemption, and featuring sub-machine granularity resource assignment within a single scheduling cycle. The first set of experiments with a variety of job type mixes, workload intensities, degrees of burstiness, preference strengths, and input inaccuracies support our hypothesis that leveraging space-time soft constraints is (a) beneficial and (b) possible to achieve.