



PDL Packet

NEWSLETTER ON PDL ACTIVITIES AND EVENTS • SPRING 2009

<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

Tashi.....	1
Director's Letter.....	2
DCO Expands.....	3
Year in Review.....	4
Recent Publications.....	5
Birth of pNFS.....	9
PDL News & Awards.....	10
Dissertations & Proposals.....	12
PLFS.....	18

PDL CONSORTIUM MEMBERS

American Power Corporation
Data Domain, Inc.
EMC Corporation
Facebook
Google
Hewlett-Packard Labs
Hitachi, Ltd.
IBM Corporation
Intel Corporation
LSI Corporation
Microsoft Research
NEC Laboratories
NetApp, Inc.
Oracle Corporation
Seagate Technology
Sun Microsystems
Symantec Corporation
VMware, Inc.

Tashi: Cloud Computing on Big Data

Adapted from paper to be presented at ACDC'09, June 19, 2009, Barcelona, Spain [1].

Digital media, pervasive sensing, web authoring, mobile computing, scientific and medical instruments, physical simulations, and virtual worlds are all delivering vast new datasets relating to every aspect of our lives. A growing fraction of this Big Data is going unused or being underexploited due to the overwhelming scale of the data involved. Effective sharing, understanding, and the use of this new wealth of raw information poses a great challenges to today's computer researchers.

In order to effectively compute on this scale, many research and development groups purchase their own racks of compute and storage servers. The goal of the Tashi project is to develop a layer of utility software that turns these raw racks of servers into easily managed cloud computers that will allow remote users to share and explore their Big Data.

Big Data applications are typically data hungry in that the quality of their results improves with the quantity of data available. Consequently, scalable computing technologies, able to accommodate the largest datasets possible, are important. Fortunately, these applications are usually disk bandwidth limited (rather than seek-limited) and exhibit extremely good parallelism. Therefore, commodity cluster hardware, when employed at scale, may be harnessed to support such large dataset applications. For example, the cluster at Intel Research Pittsburgh that is part of the OpenCirrus consortium (<http://opencirrus.org/>) consists of a modest 150 server nodes, providing more than 1000 computing cores and over 400 TB of disk storage – enough to accommodate many current Big Data problems.

continued on page 16

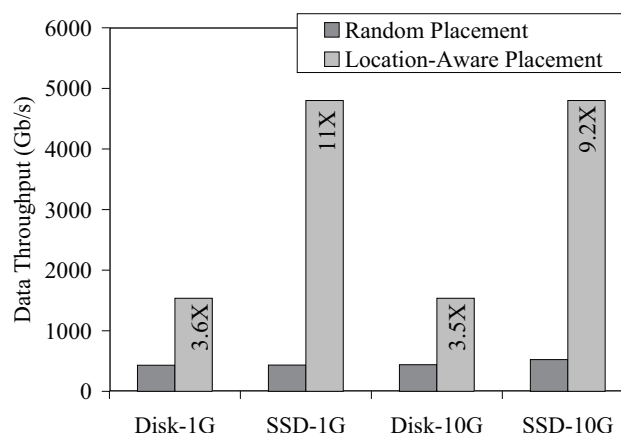


Figure 1: Performance comparison of location aware task placement with random task placement. The labels on the data bars show the performance improvement for the Location-Aware Placement relative to the Random Placement.



FROM THE DIRECTOR'S CHAIR

Greg Ganger

Hello from fabulous Pittsburgh!

With this issue, we have switched the primary PDL Packet publication date from mid-Fall to mid-Spring, with the idea being to offset publication from the PDL Retreat. As a result, just 6 months have passed since the last issue.

Still, many great things have happened on the

PDL research front. Some highlights include addition of many servers to the Data Center Observatory (DCO) for cloud computing, beta deployments of our Perspective home storage system, and new Fellowships supporting our research efforts in energy efficiency. Along the way, many students have accepted positions with PDL Consortium companies, and many papers have been published. Let me highlight a few things.

The DCO continues to be a nexus for PDL research activity, building on its long-term (6 years and counting) vision of enabling CMU research into automated cloud computing, scalable storage, and data center energy efficiency. In March, we hosted a small ceremony (discussed more in the DCO article) to commemorate the addition of the second APC power and cooling “zone”, which added capacity for another 400 servers. Over the past few months, we have planned the usage of about half of that capacity, ordering 150 new servers (over half donated by Intel) for deployment as part of two open cloud computing testbeds: the OpenCloud testbed (led by the Open Cloud Consortium) and the OpenCirrus testbed (led by Intel, Yahoo!, and HP). These testbeds will greatly enhance CMU researchers’ ability to explore new computing models, such as data-intensive scalable computing (DISC), and system support for them. To provide the virtualized cluster management, we have been actively working on the open source Tashi project, initiated jointly by PDL, Intel, and Yahoo! to create widely available software for cloud computing.

At the March event, APC announced their new APC Fellowship program, which recognizes and supports research in Data Center Efficiency. Three PDL Ph.D. students won Fellowships for their work, primarily focused on energy efficiency. The recognized activities include (1) exploiting data available from the DCO’s deep instrumentation to develop and evaluate models for dynamic measurement and adaptive control of power and cooling in data centers; (2) exploring cluster scheduling to maximize efficiency for virtualized data centers; (3) exploring alternate hardware architectures for large-scale data-intensive computations, using large arrays of low-power nodes (based on embedded CPUs and Flash) to provide performance comparable to more conventional clusters at over an order of magnitude lower energy.

Our home/consumer storage project is maturing nicely, advancing to the stage of beta deployments and user studies. The Perspective system, designed to simplify data management and sharing among the many storage-enhanced devices (e.g., DVRs, iPods, laptops), is now being used in a lounge at CMU as well as in a few PDL students’ homes. Continuing development is readying it for deployment in the homes of some non-experts, enabling in situ user studies. Our FAST 2009 paper describes Perspective, and a significant research thrust going forward focuses on security mechanisms and access control policy management for this challenging environment.

The Petascale Data Storage Institute (PDSI), led by Garth Gibson, continues to

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

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

STAFF MEMBERS

Bill Courtright 412•268•5485
(pdl executive director) wcourtright@cmu.edu
Karen Lindenfelser, 412•268•6716
(pdl business administrator) karen@ece.cmu.edu
Mike Bigrigg
Joan Digney
Adam Goode
Nitin Gupta
Doug Needham
Manish Prasad
Michael Stroucken
Spencer Whitman
Charlene Zang

GRADUATE STUDENTS

Michael Abd-El-Malek	Luca Parolini
Mukesh Agrawal	Swapnil Patil
Azim Ali	Adam Pennington
Jim Cipar	Amar Phanishayee
Debabrata Dash	Milo Polte
Janice D'Sa	Rob Reeder
Tudor Dumitras	Dmitriy Ryaboy
Anshul Gandhi	Brandon Salmon
Varun Gupta	Raja Sambasivan
Nikos Hardavellas	Karan Sanghi
James Hendricks	Hiral Shah
Shailesh Jain	Saurabh Shah
Wesley Jin	Tomer Shiran
Ryan Johnson	Zoheb Shivani
Christina Johns	Geeta Shroff
Mike Kasick	Jiri Simsa
Soila Kavulya	Shafeeq Sinnamohideen
Andrew Klosterman	Ajay Surie
Elie Krevat	Jiaqi Tan
Patrick Lanigan	Lawrence Tan
Yuan Liang	Wittawat Tantisirroj
Eugene Marinelli	Vijay Vasudevan
Michelle Mazurek	Gaurav Veda
Iulian Moraru	Matthew Wachs
Jim Newsome	Adam Wolbach
Xinghao Pan	Lin Xiao
Ippokratis Pandis	

FROM THE DIRECTOR'S CHAIR

develop the community of academic, industry, and national lab experts focused on the technology challenges faced in scaling storage systems to petascale sizes. Among other things, the third PDS Workshop, held on November 17, 2008, at Supercomputing '08, brought together this community. PDL's research in this space continues along many directions. Perhaps most excitingly, over the past 6 months, we have demonstrated effective approaches to mitigating the "incast" problem caused by data striping in high-performance networked storage. The solution involves using high-granularity timers for round-trip time estimation and timeouts in the TCP implementation. Several other research activities have also made strong progress, such as the exploration of new approaches to supporting very large-scale directories and the exploration of log-based storage for high-performance parallel checkpointing.

The Self-* Storage project continues to produce exciting new results and approaches to building more automated scalable storage systems. Our ongoing effort to build a usable system, despite limited resources, has yielded interesting approaches to creating and maintaining cluster-based storage systems with less effort. For example, we have found that virtual machines can be used to address the porting problems associated with the client-side component of most cluster-based designs (including ours); we call the approach File System Virtual Appliances (FSVAs). We are exploring dynamic metadata server scaling without complex consistency protocols. We are exploring tools for exploiting end-to-end request flow tracing to simplify performance debugging of the complex distributed systems that are cluster-based storage systems. Of course, we also continue to make progress on automation challenges, such as insulating performance among clients sharing a storage cluster and automating provisioning and tuning decisions.

Many other ongoing PDL projects are also producing cool results. For example, we have developed a new protocol (called Zzyzx) that provides unprecedented efficiency and scalability for Byzantine fault-tolerant services, providing a scheme for metadata to complement the fault-tolerant storage scheme developed last year. Extensive explorations into automated problem diagnosis in distributed systems include study of both model-based and learning-based schemes, centered around a new general architecture for instrumentation data of various types. 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.

ONGOING DCO EXPANSION

Bill Courtright & Joan Digney

On March 5, 2009 dignitaries from CMU and APC, along with members of the Carnegie Mellon University research community gathered to officially open the second of four phases of construction of the Data Center Observatory (DCO). Consistent with the base construction of the room, this addition employs APC's InfraStruXure® architecture, which fully integrates power, cooling racks, environmental monitoring, physical security and management. At the time of this writing, the total number of machines in the DCO is 466 and includes blade servers, computing nodes, and storage servers. The current power draw is 80 kW.

Greg Ganger, CMU Professor and PDL Director introduced the DCO and its

continued on page 8

YEAR IN REVIEW

May 2009

- ❖ 11th Annual PDL Spring Industry Visit Day.

April 2009

- ❖ Nikos Hardavellas has given the talk “Near-Optimal Block Placement and Replication in Distributed Caches” for several faculty candidate interviews, and will also present at the 36th ACM/IEEE Internat’l Symposium on Computer Architecture (ISCA’09) in Austin, TX in June.

March 2009

- ❖ Phase Two of the DCO was unveiled and APC announced three fellowships would be granted to PDL students (James Cipar, Luca Parolini and Vijay Vasudevan) to study Data Center Efficiency.
- ❖ Anastasia Ailamaki presented “Shore-MT: A Scalable Storage Manager for the Multicore Era” at the 12th International Conference on Extending Database Technology (EDBT2009), in Saint Petersburg, Russia.

February 2009

- ❖ Julio López presented “Data-Intensive Scalable Computing for Science” in NASA Goddard Space Flight Center’s Information Science & Technology Colloquium Series.
- ❖ Brandon Salmon presented “Perspective: Semantic Data Management for the Home” at FAST 2009 in San Francisco. He has taken a job at the storage startup, Tintri, in Silicon Valley, and will begin there after graduation.

December 2008

- ❖ Ippokratis Pandis proposed his Ph.D. research titled “DORA: A Data-ORiented database Architecture for Efficient OLTP and BI in modern computing environments.”
- ❖ Jiaqi Tan presented “SALSA: Analyzing Logs as StAte Machines” at the USENIX Workshop on Analysis of System Logs, San Diego, CA.

November 2008

- ❖ 16th Annual PDL Retreat and Workshop.
- ❖ Milo Polte discussed “Fast Log-based Concurrent Writing of Checkpoints” at the 3rd Petascale Data Storage Workshop in Austin, TX.
- ❖ Jiri Simsa presented “Comparing Performance of Solid State Devices and Mechanical Disks” at the 3rd Petascale Data Storage Workshop in Austin, TX.

October 2008

- ❖ The second phase of the Data Center Observatory (DCO) came online, providing power, cooling, and rack space for another 400 servers.
- ❖ Christos Faloutsos gave the keynote speech at the 2008 Internet Measurement Conference on “Graph Mining: Laws, Generators and Tools.”
- ❖ Nikos Hardavellas gave a tutorial on “SimFlex and ProtoFlex: Fast, Accurate, and Flexible Simulation of Multicore Systems” at the 17th International Conference on Parallel Architectures and Compilation Techniques (PACT) in Toronto, Canada.

September 2008

- ❖ Julio López gave a half-day tutorial talk on “Data-Intensive Scalable Computing Systems for Science (DISCS)” at the Mass Storage Systems and Technology conference MSST’08 in Baltimore.
- ❖ Jiaqi worked with Steve Schlosser and Lily Mummert during his internship with Intel Research Pittsburgh over the summer.
- ❖ Abdur Rehman interned at Seagate Research, Pgh. on their Terabyte Home Project led by Erik Riedel (PDL Alumni) and Sami Iren.
- ❖ Jim Cipar worked with Intel Research Pittsburgh on a fellowship researching “Tashi: Dynamic VM Placement.”
- ❖ Jure Leskovec interned at Microsoft studying the six degrees of separation theory with regard to

internet instant messaging services.

- ❖ Elie Krevat spent the summer at VMware helping to build a prototype of a virtual data center as part of VMware’s cloud computing efforts.
- ❖ Adam Wolbach completed his M.S. degree and presented his thesis research on “Improving the Deployability of Diamond.”

August 2008

- ❖ Greg Ganger presented “Performance Insulation for Shared Cluster Storage” at the HECURA/FSIO Workshop in Arlington, VA.
- ❖ Garth Gibson presented “GIGA+: Scalable Directories for Shared File Systems” at the HECURA/FSIO Workshop in Arlington, VA.

July 2008

- ❖ Rob Reeder successfully defended his Ph.D. research titled “Expandable Grids: A user interface visualization technique and a policy semantics to support fast, accurate security and privacy policy authoring.”
- ❖ Tudor Dumitraş proposed his Ph.D. research titled “Dependency-Agnostic Online Upgrade in Distributed Systems.”
- ❖ Julio López visited LANL and gave a talk on “Data-Intensive Scalable Computing Systems for Science (DISCS).”

June 2008

- ❖ Michael Abd-El-Malek proposed his Ph.D. research on File System Virtual Appliances.”
- ❖ Swapnil Patil gave an invited talk on “GIGA+: Scalable Directories for Shared File Systems” at the Conference on Scalability 2008 organized by Google in Seattle WA.

May 2008

- ❖ 10th Annual PDL Spring Industry Visit Day.
- ❖ Christos Faloutsos was the Keynote speaker at PAKDD 2008 in Osaka Japan, presenting “Graph Mining: Laws, Generators and Tools.”

A (In)Cast of Thousands: Scaling Datacenter TCP to Kiloservers and Gigabits

Vasudevan, Phanishayee, Shah, Krevat, Andersen, Ganger & Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-09-101, Feb. 2009.

This paper presents a practical solution to the problem of high-fan-in, high-bandwidth synchronized TCP workloads in datacenter Ethernets—the Incast problem. In these networks, receivers often experience a drastic reduction in throughput when simultaneously requesting data from many servers using TCP. Inbound data overfills small switch buffers, leading to TCP timeouts lasting hundreds of milliseconds. For many datacenter workloads that have a synchronization requirement (e.g., filesystem reads and parallel dataintensive queries), incast can reduce throughput by up to 90%.

Our solution for incast uses high-resolution timers in TCP to allow for microsecond-granularity timeouts. We show that this technique is effective in avoiding incast using simulation and real-world experiments. Last, we show that eliminating the minimum retransmission timeout bound is safe for all environments, including the wide-area.

Materialized Community Ground Models for Large-Scale Earthquake Simulation

Schlosser, Ryan, Taborda, López, O'Hallaron & Bielak

Supercomputing (SC'08), Austin, TX, November 2008.

Large-scale earthquake simulation requires source datasets which describe the highly heterogeneous physical characteristics of the earth in the region under simulation. Physical characteristic datasets are the first stage in a simulation pipeline which includes mesh generation, partitioning, solv-

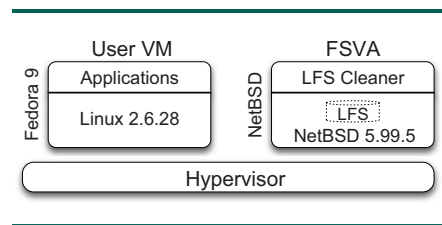
ing, and visualization. In practice, the data is produced in an ad-hoc fashion for each set of experiments, which has several significant shortcomings including lower performance, decreased repeatability and comparability, and a longer time to science, an increasingly important metric. As a solution to these problems, we propose a new approach for providing scientific data to ground motion simulations, in which ground model datasets are fully materialized into octrees stored on disk, which can be more efficiently queried (by up to two orders of magnitude) than the underlying community velocity model programs. While octrees have long been used to store spatial datasets, they have not yet been used at the scale we propose. We further propose that these datasets can be provided as a service, either over the Internet or, more likely, in a datacenter or supercomputing center in which the simulations take place. Since constructing these octrees is itself a challenge, we present three data-parallel techniques for efficiently building them, which can significantly decrease the build time from days or weeks to hours using commodity clusters. This approach typifies a broader shift toward science as a service techniques in which scientific computation and storage services become more tightly intertwined.

File System Virtual Appliances: Portable File System Implementations

Abd-El-Malek, Wachs, Cipar, Sanghi, Ganger, Gibson & Reiter

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-09-102, March 2009.

File system virtual appliances (FSVAs) address the portability headaches that plague file system (FS) developers. By packaging their FS implementation in a VM, separate from the VM that runs user applications, they can avoid the need to port the file system to each OS and OS version. A small FS-agnostic



A file system runs in a separate VM. A user continues to run their preferred OS. By decoupling the user and FS OSs, one allows users to use any OS without needing a corresponding FS port. As an example, Linux does not include a log-structured file system (LFS) implementation. But, using FSVAs, a Linux user can utilize NetBSD's LFS implementation.

proxy, maintained by the core OS developers, connects the FSVA to whatever OS the user chooses. This paper describes an FSVA design that maintains FS semantics for unmodified FS implementations and provides desired OS and virtualization features, such as a unified buffer cache and VM migration. Evaluation of prototype FSVA implementations in Linux and NetBSD, using Xen as the VMM, demonstrates that the FSVA architecture is efficient, FS-agnostic, and able to insulate file system implementations from OS differences that would otherwise require explicit porting.

FAWNdamentally Power-efficient Clusters

Vasudevan, Franklin, Andersen, Phanishayee, Tan, Kaminsky & Moraru

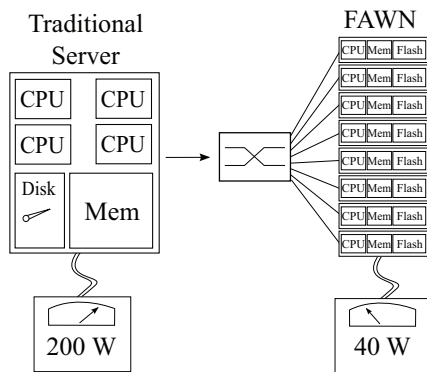
12th Workshop on Hot Topics in Operating Systems (HotOS XII). May 2009.

Power is becoming an increasingly large financial and scaling burden for computing and society. In this paper, we propose a power-efficient cluster architecture called a Fast Array of Wimpy Nodes, or FAWN. A FAWN consists of a large number of slower but efficient nodes coupled with low-power storage. Through our prelimi-

continued on page 6

RECENT PUBLICATIONS

continued from page 5



FAWN replaces a traditional cluster machine with an array of wimpy nodes, each of which consume only a few watts, thus reducing overall power draw without impairing I/O-bound workload performance.

ary evaluation, we demonstrate that a FAWN can be up to six times more efficient than traditional systems with flash storage for both seek-bound applications and I/O throughput-bound applications. Finally, we show that long-lasting, fundamental trends in the scaling of computation and energy suggest that the FAWN approach will become dominant for increasing classes of workloads.

Shore-MT: A Scalable Storage Manager for the Multicore Era

Johnson, Pandis, Hardavellas, Ailamaki & Falsafi

Proceedings of the 12th International Conference on Extending Database Technology (EDBT2009), Saint Petersburg, Russia, March 2009.

Database storage managers have long been able to efficiently handle multiple concurrent requests. Until recently, however, a computer contained only a few single-core CPUs, and therefore only a few transactions could simultaneously access the storage manager's internal structures. This allowed storage managers to use non-scalable approaches without any penalty. With the arrival of multicore chips, however, this situation is rapidly changing. More

and more threads can run in parallel, stressing the internal scalability of the storage manager. Systems optimized for high performance at a limited number of cores are not assured similarly high performance at a higher core count, because unanticipated scalability obstacles arise.

We benchmark four popular open-source storage managers (Shore, BerkeleyDB, MySQL, and PostgreSQL) on a modern multicore machine, and find that they all suffer in terms of scalability. We briefly examine the bottlenecks in the various storage engines. We then present Shore-MT, a multithreaded and highly scalable version of Shore which we developed by identifying and successively removing internal bottlenecks. When compared to other DBMS, Shore-MT exhibits superior scalability and 2-4 times higher absolute throughput than its peers. We also show that designers should favor scalability to single-thread performance, and highlight important principles for writing scalable storage engines, illustrated with real examples from the development of Shore-MT.

A Fault Model for Upgrades in Distributed Systems

Dumitraş, Kavulya & Narasimhan

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-08-115, December 2008.

Recent studies, and a large body of anecdotal evidence, suggest that upgrades are unreliable and often end in failure, causing downtime and data-loss. While this is sometimes due to software defects in the new version, most upgrade failures are the result of faults in the upgrade procedure, such as broken dependencies. In this paper, we present data on upgrade failures from three independent sources — a user study, a survey and a field study — and, through statistical cluster analysis, we construct a novel fault model for upgrades in distributed systems. We identify four

distinct types of faults: (1) simple configuration errors (e.g., typos); (2) semantic configuration errors (e.g., misunderstood effects of parameters); (3) broken environmental dependencies (e.g., incorrect libraries, port conflicts); and (4) complex procedural errors. We estimate that, on average, Type 1 faults occur in 15.2 % of upgrades, and Type 4 faults occur in 16.8 % of upgrades.

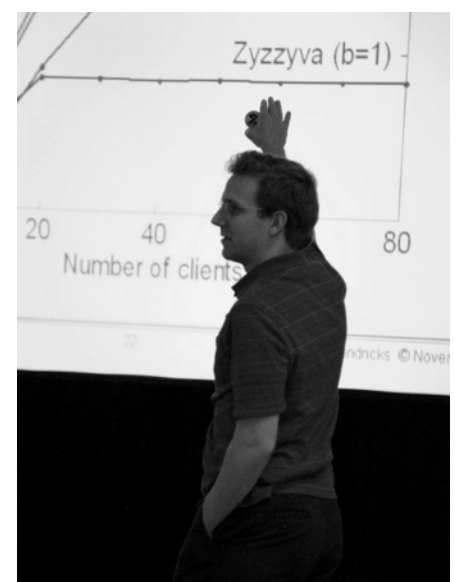
To Share Or Not To Share?

Johnson, Hardavellas, Pandis, Mancheril, Harizopoulos, Sabirli, Ailamaki & Falsafi

Proceedings of the 7th Hellenic Data Management Symposium (HDMS'08), Heraklion, Crete, Greece, July 2008.

Intuitively, aggressive work sharing among concurrent queries in a database system should always improve performance by eliminating redundant computation or data accesses. We show that, contrary to common intuition, this is not always the case in practice, especially in the highly parallel world of chip multiprocessors. As the num-

continued on page 7



James Hendricks talks about "Byzantine Fault Tolerance for Storage and Services" at the 2008 PDL Retreat & Workshop.

continued from page 6

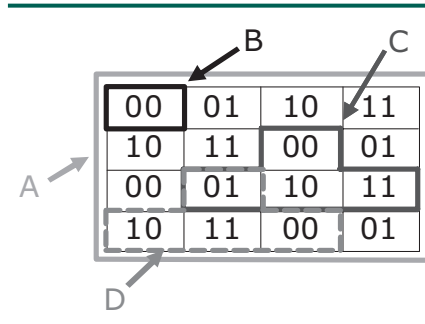
ber of cores in the system increases, a trade-off appears between exploiting work sharing opportunities and the available parallelism. To resolve the trade-off, we develop an analytical approach that predicts the effect of work sharing in multi-core systems. Database systems can use the model to determine, statically or at runtime, whether work sharing is beneficial and apply it only when appropriate. The contributions of this paper are as follows. First, we introduce and analyze the effects of the trade-off between work sharing and parallelism on database systems running complex decision-support queries. Second, we propose an intuitive and simple model that can evaluate the trade-off using real-world measurement approximations of the query execution processes. Furthermore, we integrate the model into a prototype database execution engine, and demonstrate that selective work sharing according to the model outperforms never-share static schemes by 20% on average and always-share ones by 2.5x.

Reactive NUCA: Near-Optimal Block Placement and Replication in Distributed Caches

Hardavellas, Ferdman, Falsafi & Ailamaki

Proceedings of the 36th ACM/IEEE Annual International Symposium on Computer Architecture (ISCA'09), Austin, Texas, June 2009.

Increases in on-chip communication delay and the large working sets of server and scientific workloads complicate the design of the on-chip last-level cache for multicore processors. The large working sets favor a shared cache design that maximizes the aggregate cache capacity and minimizes off-chip memory requests. At the same time, the growing on-chip communication delay favors core-private caches that replicate data to minimize delays on global wires. Recent hybrid proposals offer lower average latency than



This is an example of R-NUCA clusters and Rotational Interleaving. The array of rectangles represents the multicore processor component files. The binary numbers in the rectangles denote each tile's RID. The lines surrounding the tiles are cluster boundaries. Conceptually, R-NUCA operates on overlapping clusters of one or more files. R-NUCA introduces fixed-center clusters, which consist of the tiles logically surrounding a core. Each core defines its own fixed-center cluster. Clusters can be of various power-of-2 sizes. Clusters C and D are size-4. Cluster B is a size-1 cluster. Size-16 clusters comprise all tiles (cluster A). Data within each cluster are interleaved among the participating L2 slices, and shared among all cores participating in that cluster.

conventional designs, but they address the placement requirements of only a subset of the data accessed by the application, require complex lookup and coherence mechanisms that increase latency, or fail to scale to high core counts. In this work, we observe that the cache access patterns of a range of server and scientific workloads can be classified into distinct classes, where each class is amenable to different block placement policies. Based on this observation, we propose Reactive NUCA (R-NUCA), a distributed cache design which reacts to the class of each cache access and places blocks at the appropriate location in the cache. R-NUCA cooperates with the operating system to support intelligent placement, migration, and replication without the overhead of an explicit coherence mechanism for the on-chip last-level cache. In a range of server, scientific, and multiprogrammed workloads, R-NUCA matches the

performance of the best cache design for each workload, improving performance by 14% on average over competing designs and by 32% at best, while achieving performance within 5% of an ideal cache design.

Tashi: Location-aware Cluster Management

Kozuch, Ryan, Gass, Schlosser, O'Hallaron, Cipar, Krevat, Stroucken, López & Ganger

First Workshop on Automated Control for Datacenters and Clouds (ACDC'09), Barcelona, Spain, June 2009.

Big Data applications, those that require large data corpora either for correctness or for fidelity, are becoming increasingly prevalent. Tashi is a cluster management system designed particularly for enabling cloud computing applications to operate on repositories of Big Data. These applications are extremely scalable but also have very high resource demands. A key technique for making such applications perform well is Location-Awareness. This paper demonstrates that location-aware applications can outperform those that are not location aware by factors of 3-11 and describes two general services developed for Tashi to provide location-awareness independently of the storage system.

Fast Log-based Concurrent Writing of Checkpoints

Polte, Simsa, Tantisiriroj, Gibson, Dayal, Chainani & Uppugandla

Proceedings of the 3rd Petascale Data Storage Workshop held in conjunction with Supercomputing '08, November 17, 2008, Austin, TX.

This report describes how a file system level log-based technique can improve the write performance of many-to-one write checkpoint workload typical for high performance computations. It is shown that a simple log-based organi-

continued on page 20

DCO EXPANDS

continued from page 3

capabilities. Domenic Alcaro, APC's VP of Enterprise Sales spoke about high-density data center design and operation. During his remarks, Domenic made the announcement that three Carnegie Mellon University Ph.D. students will be awarded APC Fellowships for Data Center Efficiency Research. The APC Research Fellowships will support students with a research focus in the broad area of data center efficiency (see the news item on page 10). Receiving the fellowships are Luca Parolini, a Ph.D. student in electrical and computer engineering, and Vijay Vasudevan and James Cipar, both Ph.D. students in computer science.

Background

The first phase of the Data Center Observatory (DCO), originally conceived in 2003, went online in April of 2006. The process of planning and building the DCO has been a lengthy one. The first task was a scoping phase lasting over a year, concluding with allocation by Carnegie Mellon of just over 3,000 square feet on the lobby level of the Collaborative Innovation Center. In June of 2005, working with the University's Campus Design organization and APC, our partner for power & cooling solutions, an outside engineering firm was engaged to produce a detailed design of all mechanical systems necessary to operate the DCO (power, cooling, monitoring, flooring, fire suppression, access control, etc.). Construction began in December 2005 and was substantially complete by the end of March 2006. Teams from APC arrived to install the racks, electrical and cooling equipment, which formed the first of four InfraStruXure® zones. Following that, we began to install computers, storage servers and network switches. August of 2008 saw the process being repeated and by October, Zone 2 was operational.

The 2,000-square-foot DCO has been designed to accommodate 40 racks of computers weighing 2,000



Visitors tour the DCO on the day Zone 2 is officially opened. In the foreground, Greg Ganger (l), PDL Director, discusses the DCO's expanded facilities with Domenic Alcaro (r), VP, APC Enterprise Sales, North America.

pounds each and consuming a total of 774 kW. Two of an eventual four high-efficiency APC InfraStruXure® zone systems for powering, cooling, racking and managing equipment are now in place. The first zone houses 326 machines with 530 terabytes of storage. An additional 140 machines, including 125 dual quad-core machines, are currently being installed in the second zone. Each zone is fully instrumented to enable pinpoint monitoring of the facility.

As well as operating as a vehicle for studying data center challenges and solutions, the DCO is a shared computing and storage utility. Most of the computers are currently used by affiliated PDL and CyLab researchers for software development and distributed system experimentation, but there are also external "customers" as well. The addition of machines used by nonaffiliated people is an important step in building the shared infrastructure.

Research in the DCO

As a collaborative research facility, the DCO is extensively instrumented

and offers insight into a wide range of operational costs. We are collecting information on how administrators spend their time, how much power and cooling is required, and where those resources go (down to the individual machine level). Along with providing a platform for the live study of improving the energy efficiency of data centers, data center failure rates and the consequences of failures, and of reducing costs by sharing resources among users, the DCO also houses an increasing number of research groups using the storage resources of the facility for their own research.

The DCO is currently hosting machines dedicated to a diverse set of projects including PDL's Self-* Storage, a new storage architecture; CyLab's Biometric research project; the Datapostory — a testbed for measurement studies and evaluating distributed systems; Fingerprinting — automated problem diagnosis research; Firefly Sensor Networks; Tashi (see the article beginning on page 1) and more. Current user activities utilizing communal computing resources

continued on page 9

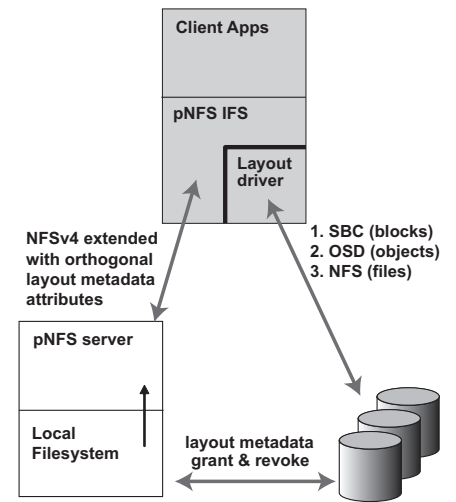
It is time to add the newest IETF file system protocol, Parallel NFS (pNFS), a component in version 4.1 of NFS to the list of descendants of PDL's late 90s project, Network-Attached Secure Disks (NASD). Also included are the PanFS and Lustre file systems dominating storage for HPC computing, and the Google File System in the cloud,

NFSv4.1 was approved in December 2008 by the architects of the IETF and forwarded to the editor for formatting into an RFC document. At this point, instead of adding the half dozen or so new pNFS metadata commands, and fixing the failure modes of an NFS server's reply cache with a powerful (that is, complex) sessions protocol, the NFS committee will likely rewrite the NFS standard. Given that the old NFS standard was already the largest IETF RFC document, ... well, lets just

say it's going to take the IETF editor a bit of time to get the document out.

Of course, a document is not much by itself. IETF processes require implementations. IBM, NetApp, Sun, EMC, LSI, Panasas, and the University of Michigan have all been developing implementations and testing interoperability of these multiple times a year.

Implementations are also not enough. The semi-official deployment strategy for pNFS is an implementation in and shipping with Linux, the operating system most commonly found on data-center compute servers most likely to deploy pNFS. To that end, some of the companies mentioned are contributing, along with Michigan, code for NFSv4.1 in Linux. This critical activity hit a milestone of its own recently. Linux



pNFS architecture.

2.6.30 started taking NFS v4.1 code into the tree! And to add to the good news, RedHat Fedora is now hosting a git tree of Linux pNFS code.

Microsoft also just announced that it is funding the team at Michigan to develop an open source NFSv4.1 client for the Windows operating system. Given that Sun has its client in the OpenSolaris tree, that will be three open source client pNFS implementations in three leading operating systems.

For more reading material, including pointers to the IETF documents, Linux git trees, and slides from an industry BOF on the topic, visit pnfs.com.

From: "Spencer Shepler" <shepler@storspeed.com>
Date: December 19, 2008 8:20:56 AM CMT-05:00
To: <nfsv4@ietf.org>
Subject: [nfsv4] NFSv4.1 I-Ds have been approved for RFC publication

If you have not been keeping your score card up to date, the IESG have finished their review and approval of the NFSv4.1 I-Ds. The IETF announcement of their approval is pending and then they will move on to the RFC editor queue for final publication.

We are DONE!

It's in writing!

DCO EXPANDS

continued from page 8

include distributed fault tolerance experiments, physical simulations (nanotech, earthquakes), data mining, intrusion detection studies, processing vast amounts of Internet trace data and a wide variety of software development and testing. The DCO will also soon host parts of two new cloud-computing research testbeds — OpenCirrus and Open Cloud.

OpenCirrus, established by Intel, HP and Yahoo!, is an open cloud-computing research testbed designed

to support research into the design, provisioning, and management of services at a global, multi-datacenter scale. The DCO will also become a part of the Open Cloud Testbed, contributing computing, networking and other resources to the members of the Open Cloud Consortium. The OCC is a newly formed group of universities that is both trying to improve the performance of storage and computing clouds spread across geographically disparate data centers and promote

open frameworks that will let clouds operated by different entities work seamlessly together. The Open Cloud Testbed is currently the only wide area cloud that utilizes wide area 10 Gb/s networks.

For more information on the DCO and to take a virtual tour, which takes a walk through the physical infrastructure of the facility, please visit the DCO website at <http://www.pdl.cmu.edu/DCO/index.html>

AWARDS & OTHER PDL NEWS

April 2009

Michelle Mazurek Named as an ECE Endowed Fellowship Winner



Michelle Mazurek, advised by Greg Ganger and Lujo Bauer, has been awarded the Lamme/Westinghouse ECE Graduate (PhD) Fellowship.

The B.G. Lamme/Westinghouse Graduate Fellowship Fund has given Carnegie Mellon an endowment valued at \$1,161,426 that will be used to provide graduate fellowships in the Department of Electrical and Computer Engineering. Westinghouse Electric established the Lamme Scholarship Fund for graduate study in electrical engineering in 1927 in memory of Westinghouse chief engineer Benjamin Garver Lamme, who died in 1924. Originally, the fund was only to be used for graduate study by Westinghouse engineers. When Westinghouse Electric was dismantled in the late 1990s, Carnegie Mellon was granted this endowment to support fellowships in electrical and computer engineering. With this gift, Westinghouse Electric has contributed more than \$3.2 million to the university since 1985.

-- with info from CMU 8.5xII News

April 2009

Dave Andersen's FAWN in Technology Review

It is now well-known that power is a big problem for operating large data centers. Companies pay huge amounts of money for the energy to power and cool their server clusters, and the negative impact on the environment caused by this energy load is surprisingly high. So can we do better? David Andersen and his team have been studying this issue, experimenting with alternative server architectures that achieve order-of-magnitude better performance per

watt of energy, at least for the kinds of web service loads that companies like Amazon and Facebook experience.

Called FAWN, for "Fast Array of Wimpy Nodes", the server design uses an array of extremely low-power CPUs (the kind used in embedded systems), flash memory, and some smart software to achieve high performance with low power — each node able to handle 700 data-lookup queries per second in under 4 watts.

Check out the nice article in Technology Review (<http://www.technologyreview.com/computing/22504/?a=f>) and, of course, FAWN has just been Slashdotted as well.

--Peter Lee's CSDiary

March 2009

University Engineers Create "YinzCam" to give Sports Fans Unique Access

Carnegie Mellon engineering faculty and their students have created a new, unique large-scale mobile wireless video service designed to enhance sports fans' experience at games. "YinzCam" is designed to help fans select and view live video feeds from unique camera angles throughout a sporting arena, according to Priya Narasimhan, associate professor of electrical and computer engineering and director of the university's Mobility Research Center. Rajeep Gandhi, a systems faculty researcher in the Electrical and Computer Engineering Department (ECE) and the Information Networking Institute, said YinzCam was a unique opportunity to apply research in the real world in a tangible, high-impact way.

Spurred by a dramatic rise in the demand for mobility services, the YinzCam gives fans the ability to obtain mobile video, real-time action replays, game-time information, statistics and player bios right from their stadium or arena seats. In collaboration with the Pittsburgh Penguins, the researchers have launched a pilot program at

Mellon Arena where hockey fans are using their wi-fi-based devices to enjoy the features of the system.

The work fits nicely with the university's newly created Mobility Research Center, where faculty and students conduct research to improve hardware and software technologies, including studies of how people work, play, shop, collaborate, and how new applications and services can change their lives.

-- CMU 8.5xII News Mar 12, 2009

March 2009

Three PDL students Awarded APC Fellowships for Data Center Efficiency Research

Three Carnegie Mellon students have been awarded APC Fellowships for Data Center Efficiency Research. Luca Parolini, a Ph.D. student in electrical and computer engineering, and Vijay Vasudevan and James Cipar, both Ph.D. students in computer science, received fellowships from APC that will cover tuition and stipends for one year. The APC Research Fellowships support Carnegie Mellon Ph.D. students with a research focus in the broad area of data center efficiency.

"I am extremely grateful for this recognition by APC because there is so much pressure for industry to cut energy consumption, and the award will support the university's ongoing research into improving data center performance," said Parolini of Padua, Italy.



continued on page 11

continued from page 10

Vasudevan said he was both excited and proud of the APC Research Data Center Fellowship award. "This is a wonderful honor and I know it will help with my research," said Vasudevan of Palo Alto, Calif. Currently, Vasudevan is building computer clusters that consume only five to six watts of electricity compared with the current industry-wide standard of 300 to 500 watts.

"Improving energy efficiency will continue to be an essential factor in the development and implementation of data center solutions having a critical affect on IT and facilities assets alike," said Robert McKernan, APC's senior vice president and president of APC North America. "The APC Fellowships for Data Center Efficiency Research will enable these Carnegie Mellon University students to research and influence key trends in the critical power and cooling industry."

-- with information from CMU News Press Release, March 4, 2009 and CMU 8.5xII News, March 5, 2009

February 2009 Polo Chau Receives Symantec Research Labs Fellowship



Polo Chau, a doctoral candidate in the Machine Learning Department in the School of Computer Science, is one of three recipients this year of the Symantec Research Labs Graduate Fellowship, awarded to promising graduate students with a demonstrated interest in solving real-world information security, storage and systems management problems. Chau's research combines the fields of machine learning and human-computer interaction to create visual and interactive graph mining systems that help analysts keep pace

with emerging threats by identifying and specifying anomalous patterns and instructing a system to detect them. He was also a recipient of the Symantec Research Labs Graduate Fellowship awarded in June 2008. His research efforts as an intern contributed to the development of Symantec's innovative reputation-based approach to malware detection. The one-year fellowship covers 100 percent of tuition and fees, along with a competitive stipend to fund Chau's ongoing research.

-- CMU 8.5xII News Feb. 26, 2009

February 2009 Two PDL Students Receive the IBM Graduate Fellowship

Congratulations to Ryan Johnson, CS and Amar Phani-shayee, CS, who have both received an IBM Graduate Fellowship.



IBM's Ph.D. Fellowship Award is an intensely competitive program which honors exceptional Ph.D. students in many academic disciplines and areas of study such as computer science and engineering, electrical and mechanical engineering, physical sciences, mathematical sciences, business sciences, etc. Focus areas of interest include technology that creates new business value, innovative software, new types of computers, and interdisciplinary projects. The IBM Ph.D. Fellowships are awarded worldwide and consist of tuition, fees, and a stipend for one nine-month academic year based on the country in which the student is studying. All IBM Ph.D. Fellows are matched with an IBM Mentor according to their technical interests, and are encouraged to participate in



an internship while completing their studies.

January 2009 Narasimhan Wins Carnegie Science Award

Priya Narasimhan, associate professor of electrical and computer engineering, has been awarded a Carnegie Science Center



Award, receiving the Emerging Female Scientist Award.

Narasimhan was recognized as a leader and innovator in developing embedded and mobile technologies. Narasimhan and her team of 15 students have developed what they call a "smart football." By installing a mini GPS unit and accelerometer inside the ball, they can plot the football's progress and landing, even under a pile of players. They've also developed a "smart glove" embedded with 15 sensors in the fingers and palm, which can help determine if a receiver has control of the ball during critical plays. The awards were announced Jan. 29 and will be presented at an awards ceremony May 9 at the Carnegie Music Hall.

-- from Carnegie Mellon News Blog, Winter 2009

January 2009 Carlos Guestrin Receives IJCAI Computers & Thought Award

Carlos Guestrin of the Computer Science & Machine Learning Departments is recipient of the 2009 IJCAI Computers and Thought Award. This award is given every two years to "outstanding young scientists in artificial intelligence." Congratulations Carlos!

The IJCAI Computers and Thought Award is presented by the Interna-

continued on page 14

DISSERTATIONS & PROPOSALS

DISSERTATION ABSTRACT:

Expandable Grids: A user interface visualization technique and a policy semantics to support fast, accurate security and privacy policy authoring

Rob Reeder

*Carnegie Mellon University SCS
Ph.D. Dissertation, July 21, 2008.*

This thesis addresses the problem of designing user interfaces to support creating, editing, and viewing security and privacy policies. Policies are declarations of who may access what under which conditions. Creating, editing, and viewing—in a word, authoring—accurate policies is essential to keeping resources both available to those who are authorized to use them and secure from those who are not. User interfaces for policy authoring can greatly affect whether policies match their authors' intentions; a bad user interface can lead to policies with many errors, while a good user interface can ensure that a policy matches its author's intentions. Traditional methods of displaying security and privacy policies in user interfaces are deficient because they place an undue burden on policy authors to interpret nuanced rules or convoluted natural language.

We introduce the Expandable Grid, a novel technique for displaying policies in a user interface. An Expandable Grid is an interactive matrix visualization designed to address the problems that traditional policy-authoring interfaces have in conveying policies to users. This thesis describes the Expandable Grid concept, then presents three pieces of work centered on the concept:

- ❖ a design, implementation, and evaluation of a system using an Expandable Grid for setting file permissions in the Microsoft Windows XP operating system;
- ❖ a description and evaluation of a file-permissions policy semantics that complements the Expandable Grid particularly well for reducing policy-authoring errors; and

❖ a design, implementation, and evaluation of a system using an Expandable Grid for displaying website privacy policies to Web users.

The evaluations of the Expandable Grid system for setting file permissions and its associated policy semantics show that the Expandable Grid can greatly improve the speed and accuracy with which policy authors complete tasks compared to traditional policy-authoring interfaces. However, the evaluation of the Expandable Grid system for displaying website privacy policies suggest some limitations of the Grid concept. We conclude that the Expandable Grid is a beneficial promising approach to policy-authoring interface design, but that it must be applied with care and tailored to each domain to which it is applied.

THESIS ABSTRACT:

Improving the Deployability of Diamond

Adam Wolbach

*Carnegie Mellon University SCS M.S.
Thesis, CMU-CS-08-158, September 2008.*

This document describes three engineering contributions made to Diamond, a system for discard-based search, to improve its portability and maintainability, and add new functionality. First, core engineering work on Diamond's RPC and content management subsystems improves the system's maintainability. Secondly, a new mechanism supports "scoping" a Diamond search through the use of external metadata sources. Scoping selects a subset of objects to perform content-based search on by executing a query on an external metadata source related to the data. After query execution, the scope is set for all subsequent searches performed by Diamond applications. The final contribution is Kimberley, a system that enables mobile application use by leveraging virtual machine technology. Kimberley separates application state from a base

virtual machine by differencing the VM before and after application customization. The much smaller application state can be carried with the user and quickly applied in a mobile setting to provision infrastructure hardware. Experiments confirm that the startup and teardown delays experienced by a Kimberley user are acceptable for mobile usage scenarios.

THESIS PROPOSAL:

DORA: A Data-ORiented database Architecture for efficient OLTP and BI in modern computing environments.

*Ippokratis Pandis, SCS
December 15, 2008*

There are two main obstacles for scalability in modern hardware for database systems. The first is the time spent on the centralized lock manager. The second is the low instruction and data locality caused by the request-oriented tuple-at-a-time execution. In order to have scalable database computing, the database systems should be re-architected around a new execution model. Such a model must reduce locking overheads and improve locality. Data-oriented execution eliminates locking overheads and improves locality by distributing the locking and data accessing services across the multicore chip.

THESIS PROPOSAL:

File System Virtual Appliances

*Michael Abd-El-Malek, ECE
June 2008*

File system virtual appliances (FS-VAs) address a major headache faced by third-party FS developers: OS version compatibility. By packaging their FS implementation in a VM, separate from the VM that runs user applications, they can avoid the need to provide an FS port for every kernel version and OS distribution. A small FS-agnostic proxy, maintained by

continued on page 13

continued from page 12

the core OS developers, connects the FSVA to whatever kernel version the user chooses. Evaluation of prototype FSVA support in Linux, using Xen as the VM platform, demonstrates that this separation can be efficient and maintain desired OS and virtualization features. Experiments with three existing FSs demonstrate that the FSVA architecture can insulate FS implementations from user OS differences that would otherwise require explicit porting changes.

**THESIS PROPOSAL:
Dependency-Agnostic Online
Upgrade in Distributed Systems**

*Tudor Dumitraş, ECE
July 2008*

Online software-upgrades are unavoidable in enterprise systems. For example, business reasons sometimes mandate switching vendors; responding to customer expectations and conforming with government regulations can require new functionality. Moreover, many enterprises can no longer afford to incur the high cost of downtime and must perform such upgrades online, without stopping their systems. Most online-upgrade techniques developed over the past 40 years rely on tracking the dependencies among the components of the system-under-upgrade in order to ensure the correctness of the system, both during and after the upgrade. Today, the benefits of dependency-tracking are reaching their limit due to the increasing complexity of configuration dependencies and to the presence of dynamic dependencies that cannot always be discovered automatically. These fundamental limitations of dependency-tracking lead to frequent upgrade failures. A 2007 survey of 50 system administrators from multiple countries (82% of whom had more than five years of experience) identified broken dependencies and altered system-behavior as the leading causes of upgrade failure, followed by bugs in

the new version, incompatibility with legacy configurations and improper packaging of the new components to be deployed. According to the survey, the average upgrade-failure rate was 8.6%, with some administrators reporting that up to 50% of upgrades had failed in their respective installations.

To improve the dependability of online software-upgrades, I propose removing the leading cause of upgrade failures — broken dependencies — by presenting an alternative to dependency-tracking. While relying on knowledge of the planned changes in data-formats and observable system-behavior between the old and new versions, this approach treats the system-under-upgrade as a black box and is, by design, guaranteed not to modify the distributed dependencies within this system. The key to achieving such a dependency-agnostic upgrade is isolating the new version of the system from the old version, to avoid sharing dependencies. I enforce the dependency-isolation by installing the new version in a parallel universe — a logically distinct collection of resources, realized either using different hardware or through virtualization — that is isolated from the universe of the old version. With this approach, a complex distributed-system upgrade can be performed as an atomic action. While it cannot prevent all possible configuration errors or upgrade-failures due to external dependencies, this approach eliminates the internal single-points-of-failure for dependency-breaking faults. Experiments conducted with a prototype implementation indicate that the response time of the system-under-upgrade is not affected during the upgrade process. More specifically, using this prototype to upgrade a three-tier web application (RUBiS) reduces the risk of downtime due to broken dependencies by over 60%, compared with two widely-used alternative techniques for online upgrades in distributed systems.

**THESIS PROPOSAL
Managing Multi-core Resources in
Database Engines**

*Ryan Johnson, SCS
June 2008*

Multi-core computing causes fundamental changes in the hardware landscape with implications for all layers of the software stack. In the past, each processor generation brought significant improvements in single-thread performance, allowing existing software to benefit directly from Moore's Law. Recently, however, power constraints and diminishing returns have pushed chip manufacturers away from aggressive single-thread architecture designs. Designers now use growing transistor budgets to increase the number of hardware contexts available per chip. For the foreseeable future, Moore's Law provides software with twice as many cores each processor generation, with only modest improvements in single-thread performance. In order to achieve peak performance with these new architectures, software must provide abundant parallelism to keep an exponentially growing number of cores busy.

Software faces a second major challenge from chip multiprocessors. In contrast with the traditional shared-nothing cluster, each node of the "cluster on a chip" must share off-chip storage capacity and bandwidth, on-chip cache hierarchies, and even processor pipelines with its neighbors. Application designers must find ways to achieve parallelism without placing undue stress on shared resources. As the degree of sharing increases with the number of cores the importance of cooperation between threads will also grow. This thesis focuses on identifying and addressing the challenges for database engines which arise from this new hardware landscape. Many database engine designs date from an era dominated by I/O and few

continued on page 14

DISSERTATIONS & PROPOSALS

continued from page 13

execution contexts, while modern machines feature huge main memories and hardware support for abundant parallelism. My work will analyze the impact of multi-core computing on database engine performance and develop approaches that extract the full potential from modern architectures.

I will show that database engines must provide abundant parallelism while carefully managing shared resources such as on-chip cache hierarchies, then demonstrate ways in which this goal can be achieved. I first identify particular areas where multi-core architectures create different challenges

than traditional systems, such as managing extreme parallelism and shared resources. I then explore how to best modify database algorithms and internal database engine design, in order to address these challenges.

AWARDS & OTHER PDL NEWS

continued from page 11

tional Joint Conferences on Artificial Intelligence (IJCAI), recognizing outstanding young scientists in artificial intelligence. It was originally funded with royalties received from the book "Computers and Thought" (edited by Edward Feigenbaum and Julian Feldman), and is currently funded by IJCAI.

December 2008

Parallel NFS now approved for RFC by the Internet Engineering Task Force (IETF)

Parallel NFS (pNFS) is a part of the NFS v4.1 standard that allows clients to access storage devices directly and in parallel. The pNFS architecture eliminates the scalability and performance issues associated with NFS servers in deployment today. This is achieved by the separation of data and metadata, and moving the metadata server out of the data path. It allows delegation of layout maps (revocable callbacks) of a file's location from server to client so that clients can send requests to storage devices (currently other NFS servers, Object Storage Devices or SCSI block storage, and backend storage protocols are extensible) without intervention on the part of the server. pNFS brings together the benefits of parallel I/O with the benefits of the ubiquitous standard for network file systems (NFS). This allows users to experience increased performance and scalability in their storage infrastructure with the added assurance that their investment

is safe and their ability to choose best-of-breed solutions remains intact.

The pNFS effort was launched by Garth Gibson and Panasas with help from Los Alamos National Lab in December 2003 based on Panasas' PanFS file system for object storage, the technology spin off of CMU's Network Attached Secure Disks project (95-99) [ASPLOS98]. It was immediately supported by EMC, NetApp and Sun, and then by IBM and a few others. Reference implementations have been built by companies such as EMC, NetApp Sun, IBM, and Panasas, and a significant team is working on open source implementations for Linux under the guidance of the Linux maintainers for NFS.

October 2008

Jim and Melanie Wed!

Jim Cipar and Melanie Wilson, married October 4, 2008 in Worcester, Massachusetts. They honeymooned in Sonoma Valley and now Jim is back hard at work preparing for the retreat. Congratulations!



October 2008

Carlos Guestrin Among Popular Science's "Brilliant 10"



Carlos Guestrin, assistant professor of machine learning and computer science, has been named as one of Popular Science's "Brilliant 10,"

the magazine's annual list of top young scientists. Dubbed "the Information Wrangler" by the magazine, Guestrin was cited for developing the Cascades algorithm, which obtains the most information with the least amount of effort. The algorithm works regardless of whether you want to determine the optimal number and placement of sensors in a water distribution system or simply the best blogs to read to get news as quickly as possible.

-- CMU 8.5x11 News Oct 16, 2008

September 2008

Perspectives Developed to Thwart Internet Eavesdropping

The growth of shared Wi-Fi and other wireless computer networks has increased the risk of eavesdropping on Internet communications, but researchers at the School of Computer Science and College of Engineering have devised a low-cost system that can

continued on page 15

continued from page 14

thwart these “Man-in-the-Middle” attacks.

The researchers - David Andersen, assistant professor of computer science, Adrian Perrig, associate professor of electrical and computer engineering and public policy, and Dan Wendlandt, a Ph.D. student in computer science - have incorporated Perspectives into an extension for the popular Mozilla Firefox v3 browser that can be downloaded free of charge at www.cs.cmu.edu/~perspectives/firefox.html.

“Perspectives provides an additional level of safety to browse the Internet,” Perrig said. “To the security conscious user, that is a significant comfort.”

-- CMU 8.5xII News Sept 4, 2008

September 2008

Welcome Morgan!

Joan, Bruce and big brother Evan welcomed Morgan Mary Georgia Digney on September 17 at 4:21 am. She weighed 6 lbs 13.5 oz and was 19.5 inches long. Looks like she came out fighting!



August 2008

Greg Ganger Earns HP Innovation Research Award

CMU's Greg Ganger was one of 33 recipients worldwide to receive a 2008 HP Innovation Research Award, which is designed to encourage open collaboration with HP labs resulting in mutually beneficial, high-impact research.

Ganger, a professor of ECE and director of the Parallel Data Lab at CMU, will collaborate with HP Labs on the

research initiative titled “Toward Scalable Self-Storage.”

HP reviewed more than 450 proposals from 200 universities in 28 countries on a range of topics including intelligent infrastructure, sustainability, information explosion, dynamic cloud services and content transformation. A key element of each award will be on-campus support for one graduate student researcher.

Ganger said the award will serve to strengthen and deepen the long-standing relationship between HP Labs' scalable storage researchers and Carnegie Mellon's Parallel Data Lab.

“We will be collaborating on our common interests in scalable, self-managing storage to tackle key challenges, including performance insulation between tenants sharing a common infrastructure and tenants with different requirements,” Ganger said.

The research will also generate a prototype that can be tested in both commercial and academic venues. “HP partners with the best and brightest in the industry and academia to drive open innovation and set the agenda for breakthrough technologies that are designed to change the world,” said Prith Banerjee, senior vice president of research at HP and director of HP Labs.

-- CMU Press Release Aug 21, 2008

August 2008

Julia Alexis Arrives!

Julia Alexis was born to John and Corley Strunk on August 17 at 3:15 pm. She weighed 6 lbs 4 oz and was 20 inches long. Good luck to John and his family as he settles into his new job with NetApp.



July 2008

Jimeng Sun Runner-up for Best SIGKDD Dissertation Award

Dr. Jimeng Sun (Ph.D. CMU-CSD 2007), received the runner-up award for the best SIGKDD, which is the premier community for data mining research. Jimeng's dissertation is on tensor and stream mining, proposing novel and efficient methods to handle streams of numerical data, as well as streams of graphs. He applied his methods on chlorine monitoring in the drinking water (joint project with Prof. Jeanne VanBriesen of CIT/CMU), on monitoring the self-star data center of PDL/CMU (with Prof. Greg Ganger and his group), and also on monitoring computer traffic (with Prof. Hui Zhang, SCS/CMU).

June 2008

Welcome Reut!

After a “long wait” Reut Shiran made her entrance into the world on June 19. She is a healthy baby girl, and weighed in at 8.3 lbs and was 20.5 inches long. Her name is pronounced Re-oot (the first syllable sounds like the “re” in the color “red”). The literal meaning of the name (in Hebrew) is “friendship”.



June 2008

Priya Narasimhan Receives Teaching Award

ECE Associate Professor Priya Narasimhan has been presented an award recognizing her teaching excellence by Eta Kappa Nu (Sigma Chapter, Carnegie Mellon). Congratulations Priya!

continued from page 1

Unfortunately, Big Data sets are also relatively immobile. With a 1 Gbps connection to the Internet, moving a 100 TB data set into or out of a cluster would require approximately 10 days. Consequently, unless the ratio of transfer bandwidth to data set size increases dramatically, computation on Big Data sets will, of necessity, be in situ.

In a cloud computing setup, tasks are assigned to a combination of connections, software and services accessed over a network. This network of servers and connections is collectively known as “the cloud.” Sites on the cloud can provide a computation-hosting framework (such as a virtual machine-hosting service), where users bring their own, custom applications to the facility to operate on the data. Resources can be accessed as the users need them. Because of the flexibility this provides, we believe that hosted computation will play a major role in the exploitation of Big Data.

To leverage the cloud computing paradigm, the Parallel Data Lab and Intel Research Pittsburgh have initiated and are actively developing an open-source, cluster-management software package called Tashi that is designed to support cloud computing applications that operate on Big Data. Currently, Tashi is in production use at the Intel OpenCirrus site, and the project is hosted by the Apache Software Foundation incubator. A key feature of



PDL graduate students (from l-r) Karan Sanghi, Shailesh Jain, Abdur Rehman Pathan and Atul Talesara at the 2008 PDL Retreat & Workshop at Nemaocolin Woodlands Resort.

Tashi is its support for location-aware computing, which can improve performance by a factor of 3-11, even in modestly-sized clusters.

Location-Aware Placement is a method by which tasks are scheduled to execute on the same node on which they consume data. Thus, data can be retrieved at the disk bandwidth rate. When tasks are not properly placed, pulling data from a distant node across the network, the data is constrained not only by the bandwidth of the network components, but by any other contending data flows (Figure 1).

Though Location-Aware Placement is effective at boosting performance in Big Data applications, it may not always be the most suitable mechanism for a particular task—software developers who understand the data flow of their application should not be forced to use any particular tool to express their program. As well, in hosting environments, the varying software installation requirements of numerous users place a management burden on the cluster administration team. A better solution is to provide users with virtual machine containers that will allow them to manage their own software installations.

Tashi Software Design

Tashi is a virtualization-based cluster-management system that provides facilities for managing virtual machines. Users of Tashi are able to create collections of virtual machines that run on the cluster’s physical resources. These virtual machine collections form “virtual clusters” that operate on the Big Data stored in the cluster. Virtual clusters may host services that are then consumed by clients, users that interact with, but do not own, the virtual machines managed by Tashi. In terms of these basic virtual machine management facilities, Tashi is similar to Amazon’s EC2 infrastructure and various other research systems. Tashi differs from these systems in its support for location-aware computing.

Parallel data access on a cluster file system is a key property of any viable Big Data storage system. Because application developers will typically consider parallel data access as well as compute parallelism very carefully when writing Big Data applications, completely abstracting differences between the file systems Tashi operates on may not only be unnecessary, it may be counter-productive. While exposing important properties of the file systems is desirable, most high-performance applications will access distributed file systems through their native interfaces, so developing an API that is common across file systems is likely not necessary.

However, performance-sensitive applications, as a class, benefit from exploiting data location information. Therefore, providing a standardized facility across file systems for accessing location information will enable applications to become location-aware with minimal programmer effort.

To take advantage of location information, Big Data applications typically rely on a location-aware runtime, which is responsible for interacting with the file system to extract location information and making task placement decisions. Therefore, any service that provides location information must be queryable not only from well-known runtime components, but from individual applications as well.

Tashi’s high-level software architecture for providing location-aware services is shown in Figure 2, which considers two different environments for location-aware applications. In part (a), an application stack executes directly on the host server node. Part (b) depicts a similar application structure, except that it is executing within a virtual machine that is located on that node. In both cases, the location-aware runtime and/or application accesses two services to determine location information: a *Data Location Service*, which

continued on page 17

continued from page 16

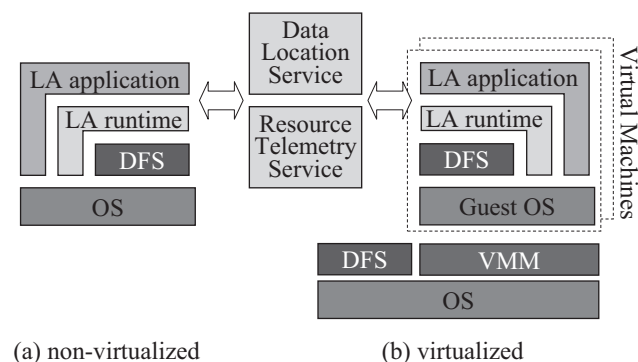


Figure 2: Software components supporting location awareness in Tashi. Location-aware (LA) applications leverage the Data Location Service and Resource Telemetry Service to obtain information regarding the location of data objects in the Distributed File System (DFS) and may execute either (a) directly on the host infrastructure or (b) inside virtual machine containers.

provides a mapping from file data blocks to storage node identifiers, and a Resource Telemetry Service, which provides information regarding the relative location of resources such as storage node identifiers.

The Data Location Service provides a mapping from file data blocks to storage node identifiers, which may be simple hostnames or IP addresses. In Tashi's current design, each Data Location Service is associated with a particular file system.

In some cases, the information returned by the Data Location Service may be sufficient for location-aware systems to place computation tasks. This interface is particularly useful in identifying notions of network distance. With a simple map of the cluster installation, the Resource Telemetry Service is able to supply useful information, such as how many switches must be traversed on communication paths from one hostId to another. This interface may also be useful to quantify metrics such as network hop count, observed latency, observed bandwidth, nominal bandwidth, etc. Virtualization can obscure some location information though, and determining which VM is "closest" to a particular

data block (and hence should host a task operating on that block) is challenging. Resolving such vagaries falls to the Resource Telemetry Service.

Tashi provides support to applications with different levels of location awareness, separating location information, resource information and scheduling decisions in different components. For example, when executing applications that are not location

aware, Tashi allocates VMs to hosts based on resource requirements and availability, and relocations may be made in response to specific requests. Location-aware applications query the Data Location Service to obtain information about the placement of the input data. The application can then use the Resource Telemetry Service to determine the distance between the input data and the initial set of VMs. Based on this information, the application can determine a task to VM assignment that reduces the overall data movement. Note that while the Resource Telemetry Service provides the right abstraction for determining appropriate placement decisions in the presence of virtualization, it does not currently provide interfaces that enable adaptation in the presence of all location changes. Determining a reasonable interface for change notification is part of our current work.

Project Status

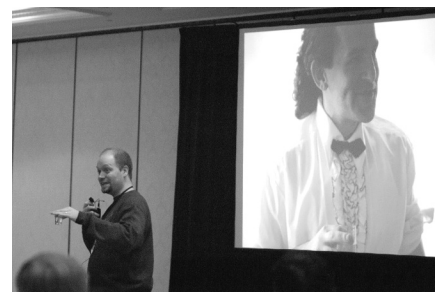
The Tashi project is currently in production use at the OpenCirrus clusters at Intel Research Pittsburgh, a cluster of approximately 150 server nodes, and will soon be operational in the PDL's Data Center Observatory with 78 server nodes.

The Tashi source code is available at the Apache Software Foundation incubator, where it is hosted. Data Location and Resource Telemetry Services have been prototyped, but those services are not yet part of the mainline implementation. Initial results are promising, however; a test application was able to leverage the prototype services to significantly improve its performance. The standalone application read through a 1 TB dataset stored in HDFS on a 28-node cluster using a random task layout in 139 minutes. Using a location-aware assignment, the same application read the same dataset in 14.5 minutes—nearly a 10X improvement.

It is clear that Tashi, enhanced with these services, will prove to be an important tool in providing Big Data services with improved performance. Future research with these tools lies in several adjacent areas, including power management and failure-resilience.

References

- [1] Tashi: Location-aware Cluster Management. Michael A. Kozuch, Michael P. Ryan, Richard Gass, Steven W. Schlosser, David O'Hallaron, James Cipar, Elie Krevat, Julio López, Michael Stroucken, Gregory R. Ganger. To appear ACDC'09, June 19, 2009, Barcelona, Spain.



PDL alumni Erik Riedel (Seagate) honors Howard Gobiuff, another former PDL member, during the second annual PDL Distinguished Alumni Award portion of the Retreat.

PARALLEL LOG-STRUCTURED FILE SYSTEM

John Bent*, Garth Gibson†, Gary Grider*, Ben McClelland*, Paul Nowoczynski‡, James Nunez*, Milo Polte†, Meghan Wingate*

Large supercomputers suffer frequent component failures. These failures are particularly problematic for many applications at LANL, and other High Performance Computing (HPC) sites, which have long run times in excess of days, weeks, and even months. Typically these applications protect themselves against failure by periodically checkpointing their progress by saving their state to persistent storage. After a failure the application can then restart from the most recent checkpoint. For many applications, saving this state into a shared single file is more convenient than striping it across multiple files. Using the shared file approach typically causes the size of writes to be small and not aligned with file system boundaries. Unfortunately this approach results in pathologically poor performance from the underlying parallel file system, which is optimized for large, aligned writes to non-shared files.

We posit that an interposition layer inserted into the existing storage stack can rearrange problematic access patterns to achieve much better performance from the underlying parallel file system. To test this, we have developed PLFS, a Parallel Log-structured File System, to act as one such layer. Measurements using PLFS on several synthetic benchmarks and

real applications at multiple HPC supercomputing centers (including Roadrunner, the faster supercomputer LANL has yet had) confirm our hypothesis: writing to the underlying parallel file system through PLFS improves checkpoint bandwidth for all tested applications and benchmarks and on all three studied parallel file systems; in some cases, bandwidth is raised by several orders of magnitude.

From a file system perspective, there are two basic checkpointing patterns: N-N and N-I. An N-N checkpoint is one in which each of N processes writes to a unique file, for a total of N files written. An N-I checkpoint differs in that all of the N processes write to a single shared file. Applications using N-N checkpoints usually write sequentially to each file, an access pattern ideally suited to parallel file systems. Conversely, applications using N-I checkpoint files typically organize the collected state of all N processes in some application specific, canonical order, often resulting in small, unaligned, interspersed writes; a pattern which derives much lower bandwidth than that achieved by N-N.

Because N-N checkpointing extracts much higher bandwidth than N-I,

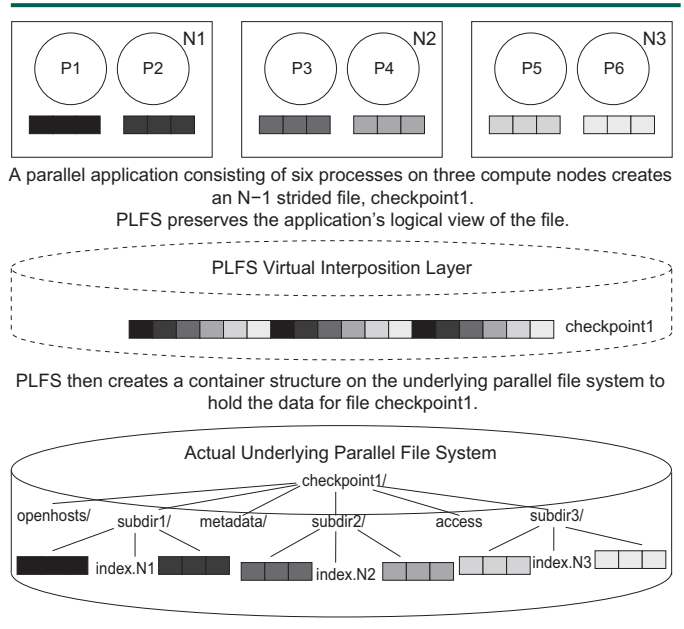


Figure 2 – PLFS Data Reorganization. This figure depicts how PLFS reorganizes an N-1 striped checkpoint file onto the underlying parallel file system. A parallel application consisting of six processes on three compute nodes is represented by the top three boxes. Each box represents a compute node, a circle is a process, and the three boxes below each process represent the state of that process. The processes create a new file on PLFS called checkpoint1, causing PLFS in turn to create a container structure on the underlying parallel file system. The container consists of a top-level directory also called checkpoint1 and several sub-directories to store the application's data. For each process opening the file, PLFS creates a data file within one of the sub-directories, it also creates one index file within that same sub-directory which is shared by all processes on a compute node. For each write, PLFS appends the data to the corresponding data file and appends a record into the appropriate index file. This record contains the length of the write, its logical offset, and a pointer to its physical offset within the data file to which it was appended. To satisfy reads, PLFS aggregates these index files to create a lookup table for the logical file. Also shown in this figure are the access file, which is used to store ownership and privilege information about the logical file, and the openhosts and metadata sub-directories which are used to cache metadata in order to improve query time (e.g. a stat call).

the obvious path to faster N-I checkpointing is for application developers to rewrite these applications to do N-N checkpointing instead. Since the bandwidth limitations of N-I are well known, some developers have done just this and have converted from an N-N to an N-I approach. Many others however continue to prefer an N-I pattern because of several advantages such as easier archiving, management,

continued on page 19

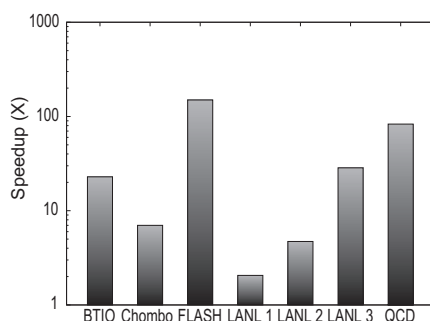


Figure 1 – Summary Of Our Results. This graph summarizes our results. The key observation here is that our technique has improved checkpoint bandwidths for all seven studied benchmarks and applications by up to several orders of magnitude.

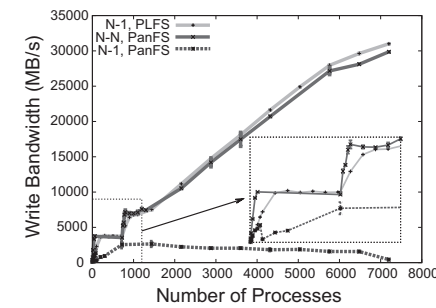
continued from page 18

visualization, and non-uniform re-start where the redistribution of the checkpoint is to a different number of processes than the number who created it. Thus, developing a method by which an N-I pattern can achieve the bandwidth of an N-N pattern while still benefiting from N-I advantages would be a tremendous boon to the HPC checkpoint-restart model.

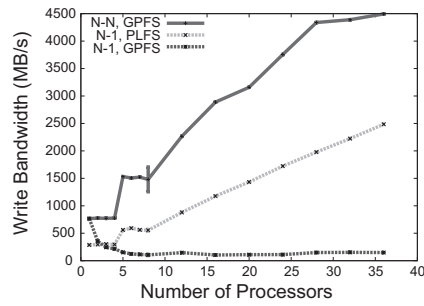
PLFS is an interposition layer which transparently rearranges an N-I checkpoint pattern into an N-N pattern and thereby decreases checkpoint time by taking advantage of the increased bandwidth achievable via an N-N pattern. The basic architecture is illustrated in Figure 2. PLFS is a virtual file system situated between the parallel application and an underlying parallel file system responsible for the actual data storage. As a virtual file system, PLFS leverages many of the services provided by the underlying parallel file system such as redundancy, high availability, and a globally distributed data store. This frees PLFS to focus on just one specialized task: rearranging application data so the N-I write pattern is better suited for the underlying parallel file system.

For every logical file created by an application, PLFS creates a container structure on the underlying parallel file system. Internally, the structure of a container is a hierarchical directory tree consisting of a single top-level directory and multiple sub-directories that are hidden from the application. PLFS constructs a logical view of a single file from this container structure such that the application sees only its logical file exactly as it expects it to be. Furthermore, this logical view is not just available to the application; existing tools such as cp, gzip, tar, grep, etc, all work on a PLFS file without any modification.

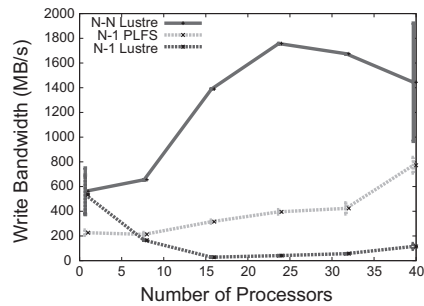
Multiple processes opening the same logical file for writing share the container although each open gets a unique data file within the container into which all of its writes are ap-



(a) MPI-IO Test on PanFS



(b) MPI-IO Test on GPFS



(c) MPI-IO Test on Lustre

Figure 3 – Experimental Results. These three graphs demonstrate the large discrepancy between achievable bandwidth and scalability using N-N and N-I checkpoint patterns on three of the major HPC parallel file systems. The green line shows how PLFS allows an N-I checkpoint to achieve most, if not all, of the bandwidth available to an N-N checkpoint.

ended. By giving each writing process in a parallel application access to a non-shared data file, PLFS converts an N-I write access pattern into a N-N write access pattern. When the process writes to the file, the write is appended to its data file and a record identifying the write is appended to an index file. Thus, PLFS preserves the advantages of an N-I pattern while managing to

extract the much higher bandwidth available to an N-N pattern.

Figures 3a, 3b, and 3c present some of the results of our study using the LANL synthetic checkpoint tool, MPI-IO Test, on three different parallel file systems, PanFS, GPFS, and Lustre. For each of these graphs, the size of each write was 47001 bytes (a small, unaligned number observed in actual applications to be particularly problematic for parallel file systems). The three lines show the bandwidth achieved by writing an N-N pattern directly to the underlying parallel file system, the bandwidth achieved by writing an N-I pattern directly to the underlying parallel file system, and the third line is the bandwidth achieved by writing an N-I pattern indirectly to the underlying parallel file system through PLFS. For all three file systems, PLFS significantly improves the bandwidth of an N-I pattern. This is particularly true for the PanFS results, which were run on LANL’s Roadrunner super-computer, which shows how PLFS achieves the full bandwidth of an N-N pattern (i.e. up to about 31 GB/s). In fact, for several of the points, an N-I pattern on PLFS actually outperforms an N-N pattern written directly to PanFS.

Many details of the PLFS architecture and our evaluation were omitted here. For more information, please refer to our Technical Report at <http://www.pdsi-scidac.org/publications/papers/plfs.pdf>. [1]

Reference

- [1] PLFS: A Checkpoint Filesystem for Parallel Applications. John Bent, Garth Gibson, Gary Grider, Ben McClelland, Paul Nowoczynski, James Nunez, Milo Polte, Meghan Wingate. LANL Technical Release LA-UR 09-02117, April 2009.

*Los Alamos National Laboratory
†Carnegie Mellon University
‡Pittsburgh Supercomputing Center

RECENT PUBLICATIONS

continued from page 7

zation can provide for substantial improvements in the write performance while retaining the convenience of a single flat file abstraction. The improvement of the write performance comes at the cost of degraded read performance however. Techniques to alleviate the read performance penalty, such as file reconstruction on the first read, are discussed.

Comparing Performance of Solid State Devices and Mechanical Disks

Polte, Simsa & Gibson

Proceedings of the 3rd Petascale Data Storage Workshop held in conjunction with Supercomputing '08, November 17, 2008, Austin, TX.

In terms of performance, solid state devices promise to be superior technology to mechanical disks. This study investigates performance of several up-to-date high-end consumer and enterprise Flash solid state devices (SSDs) and relates their performance to that of mechanical disks. For the purpose of this evaluation, the IOZone benchmark is run in single-threaded mode with varying request size and access pattern on an ext3 filesystem mounted on these devices. The price of the measured devices is then used to allow for comparison of price per performance. Measurements presented in this study offer an evaluation of cost-effectiveness of a Flash based SSD storage solution over a range of workloads. In particular, for sequential access pattern the SSDs are up to 10 times faster for reads and up to 5 times faster than the disks. For random reads, the SSDs provide up to 200x performance advantage. For random writes the SSDs provide up to 135x performance advantage. After weighting these numbers against the prices of the tested devices, we can conclude that SSDs are approaching price per performance of magnetic disks for sequential access patterns workloads and are superior technology to magnetic disks for random access patterns.

Perspective: Semantic Data Management for the Home

Salmon, Schlosser, L. Cranor & Ganger

7th USENIX Conference on File and Storage Technologies (FAST '09). Feb. 24-27, 2009. San Francisco, CA. Supercedes Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-08-105, May 2008.

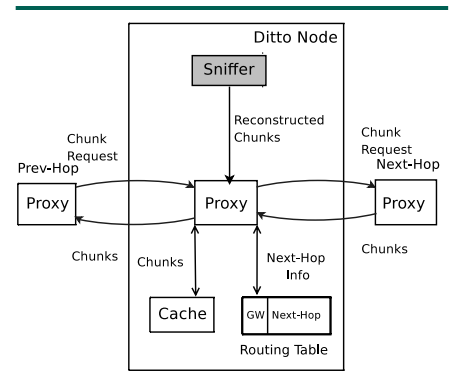
Perspective is a storage system designed for the home, with the decentralization and flexibility sought by home users and a new semantic filesystem construct, the view, to simplify management. A view is a semantic description of a set of files, specified as a query on file attributes, and the ID of the device on which they are stored. By examining and modifying the views associated with a device, a user can identify and control the files stored on it. This approach allows users to reason about what is stored where in the same way (semantic naming) as they navigate their digital content. Thus, in serving as their own administrators, users do not have to deal with a second data organization scheme (hierarchical naming) to perform replica management tasks, such as specifying redundancy to increase reliability and data partitioning to address device capacity exhaustion. Experiences with Perspective deployments and user studies confirm the efficacy of view-based data management.

Ditto - A System for Opportunistic Caching in Multi-hop Wireless Networks

Dogar, Phanishayee, Pucha, Ruwase & Andersen

The 14th Annual International Conference on Mobile Computing and Networking (Mobicom 2008). San Francisco, CA, Sept. 14-19, 2008.

This paper presents the design, implementation, and evaluation of Ditto, a system that opportunistically caches overheard data to improve subsequent

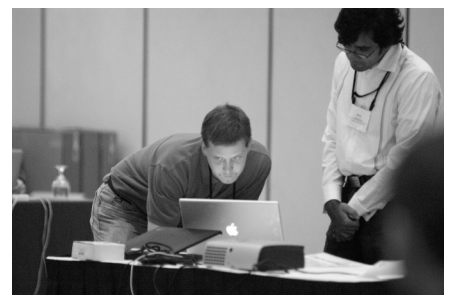


The Ditto proxy design.

transfer throughput in wireless mesh networks. While mesh networks have been proposed as a way to provide cheap, easily deployable Internet access, they must maintain high transfer throughput to be able to compete with other last-mile technologies. Unfortunately, doing so is difficult because multi-hop wireless transmissions interfere with each other, reducing the available capacity on the network. This problem is particularly severe in common gateway-based scenarios in which nearly all transmissions go through one or a few gateways from the mesh network to the Internet.

Ditto exploits on-path as well as opportunistic caching based on overhearing to improve the throughput of data transfers and to reduce load on the gateways. It uses content-based naming to provide application independent caching at the granularity of small chunks, a feature that is key to being able to cache par-

continued on page 21



Michael Stroucken and Raja Sambasivan discuss slides for Raja's talk on "Performance Diagnosis in Distributed Storage Systems" at the 2008 PDL Retreat.

continued from page 20

tially overheard data transfers. Our evaluation of Ditto shows that it can achieve significant performance gains for cached data, increasing throughput by up to 7x over simpler on-path caching schemes, and by up to an order of magnitude over no caching.

Optimal Power Allocation in Server Farms

A. Gandhi, Harchol-Balter, Das & Lefurgy

Proceedings of ACM SIGMETRICS 2009 Conference on Measurement and Modeling of Computer Systems. Seattle, WA, June 2009.

Server farms today consume more than 1.5% of the total electricity in the U.S. at a cost of nearly \$4.5 billion. Given the rising cost of energy, many industries are now seeking solutions for how to best make use of their available power. An important question which arises in this context is how to distribute available power among servers in a server farm so as to get maximum performance. By giving more power to a server, one can get higher server frequency (speed). Hence it is commonly believed that, for a given power budget, performance can be maximized by operating servers at their highest power levels. However, it is also conceivable that one might prefer to run servers

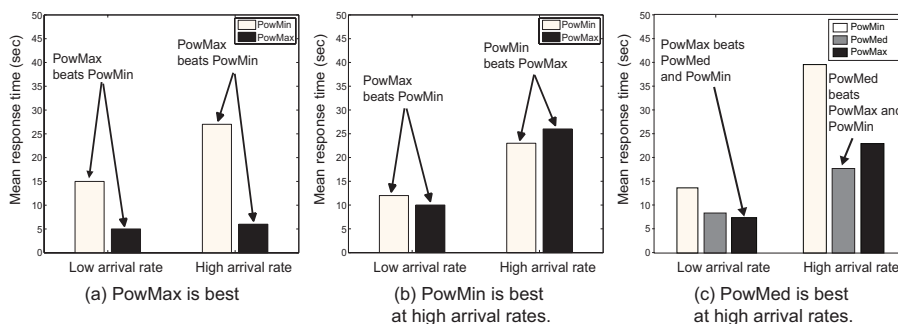
at their lowest power levels, which allows more servers to be turned on for a given power budget. To fully understand the effect of power allocation on performance in a server farm with a fixed power budget, we introduce a queueing theoretic model, which allows us to predict the optimal power allocation in a variety of scenarios. Results are verified via extensive experiments on an IBM BladeCenter. We find that the optimal power allocation varies for different scenarios. In particular, it is not always optimal to run servers at their maximum power levels. There are scenarios where it might be optimal to run servers at their lowest power levels or at some intermediate power levels. Our analysis shows that the optimal power allocation is non-obvious and depends on many factors such as the power-to-frequency relationship in the processors, the arrival rate of jobs, the maximum server frequency, the lowest attainable server frequency and the server farm configuration. Furthermore, our theoretical model allows us to explore more general settings than we can implement, including arbitrarily large server farms and different power-to-frequency curves. Importantly, we show that the optimal power allocation can significantly improve server farm performance, by a factor of typically 1.4 and as much as a factor of 5 in some cases.

Self-Adaptive Admission Control Policies for Resource-Sharing Systems

V. Gupta & Harchol-Balter

Proceedings of ACM SIGMETRICS 2009 Conference on Measurement and Modeling of Computer Systems. Seattle, WA, June 2009.

We consider the problem of admission control in resource sharing systems, such as web servers and transaction processing systems, when the job size distribution has high variability, with the aim of minimizing the mean response time. It is well known that in such resource sharing systems, as the number of tasks concurrently sharing the resource is increased, the server throughput initially increases, due to more efficient utilization of resources, but starts falling beyond a certain point, due to resource contention and thrashing. Most admission control mechanisms solve this problem by imposing a fixed upper bound on the number of concurrent transactions allowed into the system, called the Multi-Programming-Limit (MPL), and making the arrivals which find the server full queue up. Almost always, the MPL is chosen to be the point that maximizes server efficiency. In this paper we abstract such resource sharing systems as a Processor Sharing (PS) server with state-dependent service rate and a First-Come-First-Served (FCFS) queue, and we analyze the performance of this model from a queueing theoretic perspective. We start by showing that, counter to the common wisdom, the peak efficiency point is not always optimal for minimizing the mean response time. Instead, significant performance gains can be obtained by running the system at less than the peak efficiency. We provide a simple expression for the static MPL that achieves near-optimal mean response time for general distributions. Next we present two traffic-



Subset of results, showing that no single power allocation scheme is optimal. Fig. (a) depicts a scenario using DFS (Dynamic Frequency Scaling) where PowMax is optimal. Fig. (b) depicts a scenario using DVFS (Dynamic Voltage and Frequency Scaling) where PowMin is optimal at high arrival rates whereas PowMax is optimal at low arrival rates. Fig. (c) depicts a scenario using DVFS+DFS where PowMed is optimal at high arrival rates whereas PowMax is optimal at low arrival rates.

continued on page 22

RECENT PUBLICATIONS

continued from page 21

oblivious dynamic admission control policies that adjust the MPL based on the instantaneous queue length while also taking into account the variability of the job size distribution. The structure of our admission control policies is a mixture of fluid control when the number of jobs in the system is high, with a stochastic component when the system is near-empty. We show via simulations that our dynamic policies are much more robust to unknown traffic intensities and burstiness in the arrival process than imposing a static MPL.

GIGA+ : Scalable Directories for Shared File Systems

Patil & Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-08-110, August 2008.

Traditionally file system designs have envisioned directories as a means of organizing files for human viewing; that is, directories typically contain a few tens to thousands of files. Users of large, fast file systems have begun to put millions of files into single directories, for example, as simple databases. Furthermore, large-scale appli-

cations running on clusters with tens to hundreds of thousands of cores can burstily create files using all compute cores, amassing bursts of hundreds of thousands of creates or more.

In this paper, we revisit data-structures to build large file system directories that contain millions to billions of files and to quickly grow the number of files when many nodes are creating concurrently. We extend classic ideas of efficient resizable hash-tables and inconsistent client hints to a highly concurrent distributed directory service. Our techniques use a dense bitmap encoding to indicate which of the possibly created hash partitions really exist, to allow all partitions to split independently, and to correct stale client hints with multiple changes per update.

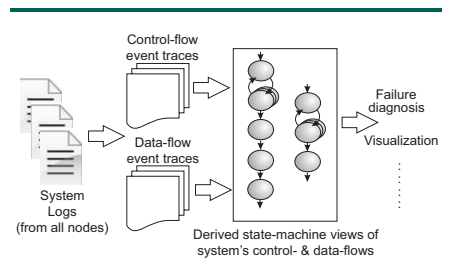
We implement our technique, GIGA+, using the FUSE user-level file system API layered on Linux ext3. We measured our prototype on a 100-node cluster using the UCAR Metarates benchmark for concurrently creating a total of 12 million files in a single directory. In a configuration of 32 servers, GIGA+ delivers scalable throughput with a peak of 8,369 file creates/second, comparable to or better than the best current file system implementations.

SALSA: Analyzing Logs as State Machines

Tan, Pan, Kavulya, Gandhi & Narasimhan

First USENIX Workshop on Analysis of System Logs (WASL). San Diego, CA. Dec 2008. Supercedes Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-08-III, Sept. 2008.

SALSA examines system logs to derive state-machine views of the system's execution, along with control-flow, data-flow models and related statistics. Exploiting SALSA's derived views and statistics, we can effectively construct



Salsa's approach.

higher-level useful analyses. We demonstrate SALSA's approach by analyzing system logs generated in a Hadoop cluster, and then illustrate SALSA's value by developing visualization and failure-diagnosis techniques, for three different Hadoop workloads, based on our derived state-machine views and statistics.

Surprising Results on Task Assignment in Server Farms with High-Variability Workloads

Harchol-Balter, Scheller-Wolf & Young

Proceedings of ACM SIGMETRICS 2009 Conference on Measurement and Modeling of Computer Systems. Seattle, WA, June 2009.

This paper investigates the performance of task assignment policies for server farms, as the variability of job sizes (service demands) approaches infinity. Our results reveal that some common wisdoms regarding task assignment are flawed. The Size-Interval-Task-Assignment policy (SITA), which assigns each server a unique size range, was heretofore thought of by some as the panacea for dealing with high-variability job-size distributions. We show SITA to be inferior to the much simpler greedy policy, Least-Work-Left (LWL), for certain common job-size distributions, including many modal, hyperexponential, and Pareto distributions. We also define regimes where SITA's performance is superior, and prove simple closed-form bounds on its performance for the above-mentioned distributions.

continued on page 23



Vijay Vasudevan presents his research on "FAWN: A Fast Array of Wimpy Nodes" at the 2008 PDL retreat

continued from page 22

Adaptive File Transfers for Diverse Environments

Pucha, Kaminsky, Andersen & Kozuch

2008 USENIX Annual Technical Conference (USENIX 2008), Boston, MA, June 22-27, 2008.

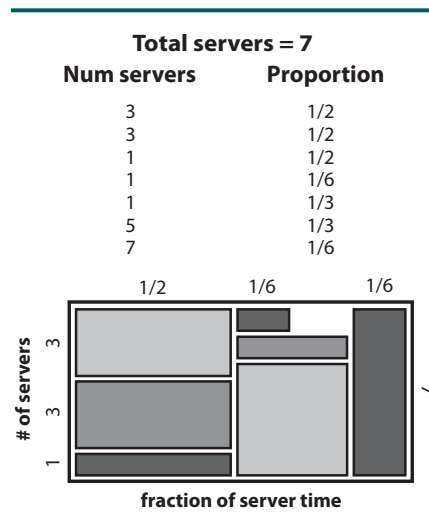
This paper presents dsync, a file transfer system that can dynamically adapt to a wide variety of environments. While many transfer systems work well in their specialized context, their performance comes at the cost of generality, and they perform poorly when used elsewhere. In contrast, dsync adapts to its environment by intelligently determining which of its available resources is the best to use at any given time. The resources sync can draw from include the sender, the local disk, and network peers. While combining these resources may appear easy, in practice it is difficult because these resources may have widely different performance or contend with each other. In particular, the paper presents a novel mechanism that enables dsync to aggressively search the receiver's local disk for useful data without interfering with concurrent network transfers. Our evaluation on several workloads in various network environments shows that dsync outperforms existing systems by a factor of 1.4 to 5 in one-to-one and one-to-many transfers.

Co-Scheduling of Disk Head Time in Cluster-based Storage

Wachs & Ganger

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-08-113, October 2008.

Disk timeslicing is a promising technique for storage performance insulation. To work with cluster-based storage, however, timeslices associated with striped data must be co-scheduled on the corresponding servers. This report describes algorithms for determining global timeslice schedules and mecha-



Example problem instance and solution. Above is an example input to the scheduling algorithm; below is one possible solution. Rectangles correspond to workloads, with their height corresponding to the number of servers and their vertical location corresponding to which servers to use; their width corresponds to share of time and their horizontal location corresponds to what span of time during which their timeslices are scheduled. The enclosing rectangle represents a single round in the cluster; the schedule is repeated indefinitely.

nisms for coordinating the independent server activities. Experiments with a prototype show that, combined, they can provide performance insulation for workloads sharing a storage cluster—each workload realizes a configured minimum efficiency within its timeslices regardless of the activities of the other workloads.

Ganesha: Black-Box Fault Diagnosis for MapReduce Systems

Pan, Tan, Kavulya, Gandhi & Narasimhan

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-08-112, October 2008.

Ganesha aims to diagnose faults transparently in MapReduce systems, by analyzing OS-level metrics alone. Ganesha's approach is based on peer-symmetry under fault-free conditions, and can diagnose faults that manifest

asymmetrically at nodes within a MapReduce system. While our training is performed on smaller Hadoop clusters and for specific workloads, our approach allows us to diagnose faults in larger Hadoop clusters and for unencountered workloads. We also candidly highlight faults that escape Ganesha's black-box diagnosis.

FAWN: A Fast Array of Wimpy Nodes

Andersen, Franklin, Phanishayee, Tan & Vasudevan

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-08-108, May 2008.

This paper introduces the FAWN—Fast Array of Wimpy Nodes—cluster architecture for providing fast, scalable, and power-efficient key-value storage. A FAWN links together a large number of tiny nodes built using embedded processors and small amounts (2–16GB) of flash memory into an ensemble capable of handling 700 queries per second per node, while consuming fewer than 6 watts of power per node. We have designed and implemented a clustered key-value storage system, FAWN-DHT, that runs atop these nodes. Nodes in FAWN-DHT use a specialized log-like back-end hash-based database to ensure that the system can absorb the large write workload imposed by frequent node arrivals and departures. FAWN uses a two-level cache hierarchy to ensure that imbalanced workloads cannot create hot-spots on one or a few wimpy nodes that impair the system's ability to service queries at its guaranteed rate. Our evaluation of a small-scale FAWN cluster and several candidate FAWN node systems suggest that FAWN can be a practical approach to building large-scale storage for seek-intensive workloads. Our further analysis indicates that a FAWN cluster is cost-competitive with other approaches (e.g., DRAM, multitudes

continued on page 24

continued from page 23

of magnetic disks, solid-state disk) to providing high query rates, while consuming 3-10x less power.

Data-intensive File Systems for Internet Services: A rose by any other name...

Tantisiriroj, Patil & Gibson

Carnegie Mellon University Parallel Data Lab Technical Report CMU-PDL-08-114, October 2008.

Data-intensive distributed file systems are emerging as a key component of large scale Internet services and cloud computing platforms. They are designed from the ground up and are tuned for specific application workloads. Leading examples, such as the Google File System, Hadoop distributed file system (HDFS) and Amazon S3, are defining this new purpose-built paradigm. It is tempting to classify file systems for large clusters into two disjoint categories, those for Internet services and those for high performance computing.

In this paper we compare and contrast parallel file systems, developed for high performance computing, and data-intensive distributed file

systems, developed for Internet services. Using PVFS as a representative for parallel file systems and HDFS as a representative for Internet services file systems, we configure a parallel file system into a data-intensive Internet services stack, Hadoop, and test performance with microbenchmarks and macrobenchmarks running on a 4,000 core Internet services cluster, Yahoo!'s M45.

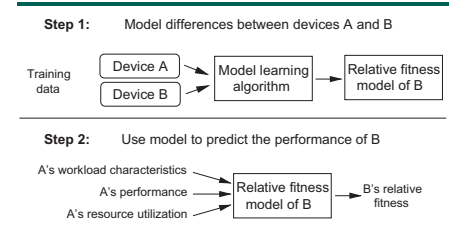
Once a number of configuration issues such as stripe unit sizes and application buffering sizes are dealt with, issues of replication, data layout and data-guided function shipping are found to be different, but supportable in parallel file systems. Performance of Hadoop applications storing data in an appropriately configured PVFS are comparable to those using a purpose built HDFS.

Relative Fitness Modeling

Mesnier, Wachs, Sambasivan, Zheng & Ganger

Research Highlights, Communications of the ACM (Vol. 52, No. 4, pg 91-96). April, 2009. ACM.

Relative fitness is a new approach to modeling the performance of storage devices (e.g., disks and RAID arrays). In contrast to a conventional model, which predicts the performance of an application's I/O on a given device, a relative fitness model predicts performance differences between devices. The result is significantly more accurate predictions.



Using sample workloads, a model learns to predict how the performance of a workload changes between two devices (A and B). To predict the performance of a new workload on B, the workload characteristics, performance, and resource utilization (as measure on device A) are input into the model of B. The prediction is a performance scaling factor, which we refer to as B's "relative fitness."



PDL Workshop and Retreat 2008.