

Asim Kadav

Curriculum Vitae

Address: 1210, W. Dayton St., Madison, WI 53705.
Phone: (608) 698 5025
Email: kadav@cs.wisc.edu
WWW: <http://cs.wisc.edu/~kadav>

RESEARCH INTERESTS

Primary Interests: Operating Systems (Device Drivers, Software Reliability, File Systems), Distributed Systems (Storage, Virtual Machines).
Secondary Interests: Programming Languages (Static Analysis, Symbolic Execution), Software Engineering.

RESEARCH SUMMARY

My dissertation research addresses the challenges arising from hardware and OS complexity in device access. My dissertation provides three major contributions over existing OS research: 1) Broad survey of the breadth of device driver code, 2) *First* research consideration of reliability in device drivers against hardware failures and, 3) Improving device downtime latency by introducing checkpoint-restore in modern device drivers.

EDUCATION

2007 - 2013	PhD in Computer Science. <i>Understanding and Improving Device Access Complexity</i> PhD Minor in Human Factors in Engineering Advisor: Prof. Michael M. Swift	University of Wisconsin-Madison
2007 - 2010	Masters with <i>Thesis</i> . <i>Live Migration of Direct-Access Devices</i>	University of Wisconsin-Madison
2000 - 2004	Bachelors of Computer Science. <i>with distinction</i>	National Institute of Technology, Nagpur, India

HONORS AND AWARDS

Best Paper Award, Hot Storage 2009.

Fast Tracked paper, WIOV 2008.

Summer Research Fellowship, Dept. of Computer Sciences, University of Wisconsin, May 2008.

Travel Grants for OSDI 2008-12, SOSP 2009 and ASPLOS 2012.

Performance award from IBM Systems and Technology Labs (2007) and Oracle Corporation (2005).

Top 0.1% across India in Mathematics and Hindi in Class X Examinations.

ARTICLES ON RESEARCH

“Trusting the Hardware too much” in Linux Weekly News (LWN), Feb 2012 on Carburizer (SOSP '09).
<http://lwn.net/Articles/479653/>

JOURNAL PUBLICATIONS

Mahesh Balakrishnan, **Asim Kadav**, Vijayan Prabhakaran, Dahlia Malkhi. Differential RAID: Rethinking RAID for SSD Reliability. *Proceedings of ACM Transactions on Storage*, Volume 6 Issue 2, July 2010. **TOS'10**

Asim Kadav and Michael M. Swift. Live Migration of Direct-Access Devices. *Proceedings of ACM SIGOPS Operating Systems Review*, Volume 43, Issue 3. July 2009. **OSR'09**

CONFERENCE PUBLICATIONS

Asim Kadav, Matthew J. Renzelmann, Michael M. Swift. Fine-Grained Fault Tolerance using Device Checkpoints. *Proceedings of Architectural Support for Programming Languages and Operating Systems*, Houston, TX, March 2013. (To appear) **ASPLOS '13**

Matthew J. Renzelmann, **Asim Kadav**, Michael M. Swift. SymDrive: Testing Drivers Without Devices. *Proceedings of Symposium on Operating Systems Design and Implementation*, Hollywood, CA, October 2012. **OSDI '12**

Asim Kadav, Michael M. Swift. Understanding Modern Device Drivers. *Proceedings of Architectural Support for Programming Languages and Operating Systems* London, UK, March 2012. **ASPLOS '12**

Mahesh Balakrishnan, **Asim Kadav**, Vijayan Prabhakaran, Dahlia Malkhi. Differential RAID: Rethinking RAID for SSD Reliability. *Proceedings of Fifth ACM European Conference on Computer Systems*, Paris, France, April 2010. **EUROSYS '10**

Asim Kadav, Matthew J. Renzelmann, Michael M. Swift. Tolerating Hardware Device Failures in Software. *Proceedings of ACM Symposium on Operating System Principles*, Big Sky, MT, October 2009. Ranked 4/23 of all accepted papers. **SOSP '09**

WORKSHOP PUBLICATIONS

Asim Kadav, Mahesh Balakrishnan, Vijayan Prabhakaran, Dahlia Malkhi. Differential RAID: Rethinking RAID for SSD Reliability. *Proceedings of Workshop on Hot Topics in Storage and File Systems*. **Best Paper Award**. Re-appeared in ACM Operating Systems Review, Volume 44 Issue 1. **HotStorage '09**

Asim Kadav and Michael M. Swift. Live Migration of Direct-Access Devices. *Proceedings of Workshop on I/O Virtualization - 2008*, co-located with OSDI 2008. Forwarded as one of the top papers to Operating Systems Review Volume 43, Issue 3. **WIOV '08**

PATENT APPLICATIONS

B. Fan, A. Kadav, E. Nightingale, J. Elson, R. Rashid, J. Mickens. Block-level Access to Parallel Storage.

OTHER PUBLICATIONS

Asim Kadav, Rathijit Sen, Michael M. Swift, Mercurial Caches: OS Support for Energy Proportional DRAM. Poster at *USENIX Operating System Design and Implementation (OSDI '12)*, Hollywood, CA, October 2012.

Asim Kadav, Chelsea Wanta, Kyung Lee, Claire Wu, Enid Montague. Improving the online video chat experience. *Proceedings of International Conference on Human-Computer Interaction*, Orlando, FL, July 2011, (HCII-14). Winner of Qualcomm Wireless Innovation Prize, 2011 and UbiComp Best Narration Award, 2011.

Matthew J. Renzelmann, **Asim Kadav**, Michael M. Swift, Testing Device Drivers without Hardware. Poster at *USENIX Operating System Design and Implementation (OSDI '10)*, Vancouver, BC, October 2010.

Asim Kadav and Mehul M. Joshi, Setting up a multicluster environment using General Parallel File System, IBM DeveloperWorks, May 2007.

TALKS

“Understanding Modern Device Drivers” at *Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '12*, London, UK. March 2012.

“ApDrive and ThinCloud” at Microsoft Research - Redmond, End of internship talk. August 2011.

“Tolerating Hardware Device Failures in Software” at *Symposium on Operating System Principles, SOSP '09*, Big Sky, MT. October 2009.

“Differential RAID: Rethinking RAID for SSD reliability” at Microsoft Research - Silicon Valley Center, End of internship talk. August 2009.

“Live Migration of Direct-Access Devices” at *Workshop on I/O Virtualization - 2008, WIOV'08*, co-located with OSDI 2008. December 2008.

SERVICE

External Review Committee: SIGMETRICS 2010, USENIX ATC 2011.

Shadow Program Committee: SOSP 2009.

WORK EXPERIENCE

Microsoft Research, Redmond, WA

June- Aug 2011 & Jan- Feb 2012

Research Intern with Ed Nightingale, Jeremy Elson and James Mickens

Developed *ThinCloud*, a system that provides fast block access to Flat DataCenter Storage (FDS) and demonstrates that desktop workloads can co-exist with big data workloads. My responsibilities included designing and implementing the kernel interface for FDS and a user-space client that implements striping for fast I/O and recovery. I also wrote simple benchmarks that demonstrate the performance and correctness of the system.

Microsoft Research Silicon Valley Center, Mountain View, CA

June-Aug 2009

Research Intern with Mahesh Balakrishnan, Vijayan Prabhakaran and Dahlia Malkhi

Designed and implemented a prototype of Differential RAID system in the Windows 7 volume manager.

IBM Systems and Technology Labs, Pune, India
Systems Software Engineer

Feb 2006 - July 2007

Member of India development team of IBM's parallel clustered file system product- GPFS (General Parallel File System) and was involved in developing the product for 64 bit Solaris SPARC V9.

Oracle Corporation, Bangalore, India
Technical Analyst

June 2004 - Feb 2006

Worked on database solutions including fault-tolerant systems, resource monitoring, database cloning, data re-organization, performance tuning and query optimization.

TEACHING EXPERIENCE

Guest Lectures on Distributed Systems and Memory Management at IBM [June 2006] and on Database Fundamentals for Quality Team at Oracle Corporation. [June 2005]

Teaching assistant at University of Wisconsin for CS 302: Introduction to Object-Oriented Programming. Supervised bi-weekly labs, created assignment, held office hours and graded homeworks and exams. [Fall 2007]

Teaching assistant at University of Wisconsin for CS 552: Introduction to Computer Architecture. Scribed lecture notes, held office hours, created and graded homework solutions, answered questions over email.[Spring 2008]

RESEARCH EXPERIENCE

Fine-Grained Fault Tolerance [ASPLOS '13]

Sept 2011 - Present

Existing driver recovery mechanisms suffer from overheads of full device restart and incomplete recovery due to proprietary device semantics. I have developed a novel device checkpointing mechanism for drivers that re-uses existing suspend/resume code in drivers. I have also developed a principled way to refactor existing drivers to export the checkpoint-restore interface. I demonstrate its utility by building Fine-Grained Fault Tolerance (FGFT), a system that executes driver entry points as transactions and uses software-fault isolation to prevent corruption and detect failures. FGFT provides very fast recovery, averaging only $20\mu s$, and is able to recover from a range of driver failures.

ThinCloud: An Elastic, High Performance Desktop File System

June 2011 - Present

Thin Cloud is a new distributed file system for desktop applications. Thin Cloud provides the scalability, fault tolerance, and sequential I/O performance of a cloud-based file system but exports a standard Win32 interface, and it handles small, random I/Os efficiently. Thin Cloud provides excellent performance to unmodified desktop applications with up to 1200 MB/s of throughput, and it improves the performance of I/O bound applications by 2x-10x. Thin Cloud can coexist alongside "big data" applications, and by using a single cloud for both desktop applications and business-scale applications, IT departments can consolidate hardware and centralize administrative tasks.

Understanding Modern Device Drivers [ASPLOS '12]

Nov 2010 - March 2012

Modern research offers a limited view of what drivers can do, mostly depicting bugs and broken code which are hardly representative of the almost 5 million lines of code (in the case of Linux). In this project, I study the form, function and abstractions of all device drivers. I find that many assumptions

made by driver research do not apply to all drivers. For example, I find that at least 42% of drivers have code that is not captured by a class definition and 14% of drivers do significant computation over data. We also find that 8% of all driver code is similar to code elsewhere, and may be removed with new abstractions or libraries.

SymDrive: Testing Device Drivers without Hardware [OSDI '12]

Feb 2010 - Oct 2012

I helped develop *SymDrive*, a system to test device drivers using symbolic execution. Symbolic execution provides the ability to test a driver without requiring any of the device hardware or chipset the driver supports. Symdrive uses static analysis to mitigate the drawbacks of symbolic execution such as state space explosion. The system also incorporates a specification framework that allows the developer to test a set of conditions each time the driver interacts with the kernel. This allows for thorough testing of drivers rather than just bug-finding.

Differential RAID: Rethinking RAID for SSD Reliability [Eurosys '10]

June 2009 - August 2009

At Microsoft Research, Silicon Valley, I worked on a RAID prototype designed for SSDs. Due to their wear-out characteristics, SSDs do not work reliably with RAID-5. This happens because RAID-5 causes all drive constituents to wear out at the same time reducing reliability. I helped develop Differential RAID, which uses parity to unevenly wear out individual SSDs in a RAID array, thereby increasing overall system reliability.

Tolerating Hardware Device Failures in Software [SOSP '09]

Oct 2008 - Oct 2009

In this project, I built *Carburizer*, a system to detect hardware dependence bugs in driver code and automatically fix them. Hardware dependence bugs are those where the software assumes correct hardware behavior and uses them unsafely in critical control and data paths. Carburizer detects and automatically fixes 1000 bugs in Linux driver code relating to livelocks, panics and unsafe memory references. Additionally, Carburizer also fixes missing error reporting, interrupt issues and provides for automatic recovery in case of hardware failures.

Live Migration of Direct-Access Devices [WIOV '08] [OSR '09]

May 2008 - Oct 2008

This project aimed at providing migration support for direct-access devices which is currently not possible. Existing solutions require one to either detach the device pre-migration or transfer all I/O to a virtual device. We designed and implemented shadow driver migration that successfully achieved live migration for directly attached network devices and is extensible to any class of devices.