



About Google

Delivering Google's products to our users requires us to develop technology across the entire systems stack, from OS device drivers all the way up to multi-site software systems that run on hundreds of thousands of computers. We design, build and operate warehouse-scale computer systems that are deployed across the globe. We construct the networks that underpin and connect them at a scale few others can match. We build storage systems that scale to exabytes, approach the performance of RAM, and never lose a byte. We use large-scale distributed systems to support code analysis and testing, making our developers some of the most productive in the industry.

Systems researchers at Google work in several modes.

- Many Googlers with research backgrounds thrive in teams that build and operate large, production-quality systems. These teams often follow a “hybrid research model” (see Google's Hybrid Approach to Research, Alfred Spector, Peter Norvig, and Slav Petrov. CACM July 2012) that deliberately blurs the line between research and engineering. This hybrid approach enables us to conduct experiments at a nearly-unprecedented scale for research projects, and tackle problems that few others can.
- Some systems Googlers work in the Google Research team, whose focus is primarily on machine learning and the systems that enable it. Their research activities span all layers of the stack from hardware design, through distributed runtimes to programming abstractions, feeding into Google's next generation ML supercomputer infrastructure.
- SystemsResearch@Google is Google's newest systems research team led by David Culler, former EECS Chair of UC Berkeley, and Hank Levy, former Head of the Allen School for Computer Science & Engineering at UW. The new team, embedded within Google's software and hardware infrastructure organization, has a mission to shape the future of hyperscale systems design.

Google and Googlers are deeply involved in the operating systems, distributed systems, and networking research communities, in our work at Google, with our publications, and in our service to the research community. We write and publish research papers to share what we have learned, and because peer feedback and interaction helps us build better systems that benefit everybody. For example, at SOSP 2021, 4 papers have Googler authors, 3 Googlers are on the Program Committee, and Google is a Sponsor of the conference.

Over the years, Googlers have been program chairs or co-chairs for many systems conferences (people in italics have moved on to other fascinating companies):

Google at SOSP 2021

HotNets: David Wetherall (2002), Steve Gribble (2008), *Dina Papagiannaki* (2018), Sylvia Ratnasamy (2019)

SIGCOMM: Christophe Diot (2000), David Wetherall (2003), Nina Taft (2007), *Dina Papagiannaki* (2009), Luigi Rizzo (2009), Jeff Mogul (2011), Arvind Krishnamurthy and Sylvia Ratnasamy (2014), Amin Vahdat (2016), Aditya Akella and Nandita Dukkupati (2021)

IMC: *Dina Papagiannaki* (2008), Nina Taft (2014), *Vivek Pai* (2015)

CoNEXT: Christophe Diot (2006), *Dina Papagiannaki* (2013), Luigi Rizzo (2014)

SOSP: Hank Levy (1997), John Wilkes (1999), Mike Dahlin (2013), Steve Hand (2015)

OSDI: David Culler (2002), Eric Brewer (2004), *Brian Bershad* and Jeff Mogul (2006), Brad Chen (2010), Amin Vahdat (2012), Hank Levy (2014), Kim Keeton (2016)

NSDI: Amin Vahdat and David Wetherall (2005), Mike Dahlin (2008), Sylvia Ratnasamy (2011), Steve Gribble (2012), Jeff Mogul (2013), Paul Barham and Arvind Krishnamurthy (2015), *Rebecca Isaacs* (2016), Aditya Akella and *Jon Howell* (2017)

EuroSys: Steve Hand (2008), John Wilkes and *Rebecca Isaacs* (2009), Kim Keeton (2016)

HotOS: Hank Levy (1995), Jeff Mogul (1997), *Matt Welsh* (2011), Petros Maniatis (2013)

FAST: Kim Keeton (2010), John Wilkes (2011), Florentina Popovici (2017), Arif Merchant (2019), Brent Welch (2020)

USENIX ATC: Jeff Mogul (Winter 1994), Wilson Hsieh (2012)

Papers at SOSP 2021 with Googler authors (in blue):

- **Snowboard: Finding Kernel Concurrency Bugs through Systematic Inter-thread Communication Analysis.** Sishuai Gong, [Deniz Altınbüken](#), Pedro Fonseca, and [Petros Maniatis](#).
- **ghOSt: Fast & Flexible User-Space Delegation of Linux Scheduling.** [Jack Tigar Humphries](#), [Neel Natu](#), [Ashwin Chaugule](#), [Ofir Weisse](#), [Barret Rhoden](#), [Josh Don](#), [Luigi Rizzo](#), [Oleg Rombakh](#), [Paul Turner](#), and Christos Kozyrakis.
- **Bladerunner: Stream Processing at Scale for a Live View of Backend Data Mutations at the Edge.** Jeff Barber, Ximing Yu, Laney Kuenzel Zamore, Shie Erlich, Jerry Lin, Vahid Jazayeri, [Tony Savor](#), and Michael Stumm.
- **WineFS: a hugepage-aware file system for persistent memory that ages gracefully.** Rohan Kadekodi, Saurabh Kadekodi, Soujanya Ponnappalli, [Harshad Shirwadkar](#), Greg Ganger, Aasheesh Kolli, and Vijay Chidambaram.

SIGOPS hall-of-fame papers with Googler authors:

(a small selection)

- *The multikernel: a new OS architecture for scalable multicore systems.* Andrew Baumann, Paul Barham, Pierre-Evariste Dagand, Tim Harris, Rebecca Isaacs, Simon Peter, Timothy Roscoe, Adrian Schüpbach, and Akhilesh Singhanian. SOSP 2009.
- *DryadLINQ: A System for General-Purpose Distributed Data-Parallel Computing Using a High-Level Language.* Yuan Yu, Michael Isard, Dennis Fetterly, Mihai Budiu, Úlfar Erlingsson, Pradeep Kumar Gunda, and Jon Currey. OSDI 2008
- *The Chubby lock service for loosely-coupled distributed systems.* Mike Burrows. OSDI 2006.
- *Bigtable: A Distributed Storage System for Structured Data.* Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. OSDI 2006
- *MapReduce: simplified data processing on large clusters.* Jeffrey Dean and Sanjay Ghemawat. OSDI 2004.
- *The Google File System.* Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. SOSP 2003.
- *Safe Kernel Extensions Without Run-Time Checking.* George C. Necula and Peter Lee. OSDI 1996.
- *A Logic of Authentication.* Michael Burrows, Martin Abadi, and Roger Needham. ACM TOCS 8(1):18-36, February 1990.
- *Reflections on Trusting Trust.* Ken Thompson. CACM 27(8), Aug 1984.
- *A Case for Redundant Arrays of Inexpensive Disks (RAID).* David A. Patterson, Garth Gibson, Randy H. Katz. SIGMOD 1988.
- *A fast file system for UNIX.* Marshall K. McKusick, William N. Joy, Samuel J. Leffler, and Robert S. Fabry. ACM TOCS 2(3), August 1984.
- *Experience with processes and monitors in Mesa.* Butler W. Lampson and David D. Redell. SOSP 1979.
- *The UNIX Time-Sharing System.* Dennis M. Ritchie and Ken Thompson. SOSP 1973.

Google at SOSP 2021

Jobs at Google

Want to join us? We're actively looking for people to work on networking, distributed systems, kernel and systems programming, and algorithms, amongst many other topics. At SOSP 2021, please find us on Slack on **#sponsors-google**, send one of us a direct message via Slack, or drop us an email at systems-phd-students@google.com if you want to chat with our researchers about the projects and opportunities at Google and how we solve interesting problems for billions of people. If you have particular interest in the new SystemResearch@Google team, contact systemsresearch@google.com.

We often hire SOSP attendees for internships and full-time jobs; please apply at careers.google.com. For internships, apply **ASAP** at careers.google.com/students – don't wait until 2022! You can also register your interest in potential future Google jobs at <http://goo.gle/sys-confs> -- note that this form is not itself a job application.

You can find more information about networking-related jobs at g.co/research/networks; note that our "networking" teams hire a lot of distributed systems and computer architecture people.

We'll be hosting an Ask Me Anything session at SOSP on Tuesday, October 26, 2:30pm - 3:00pm PDT (UTC-07) – everyone is welcome! We'll post meeting details on Slack **#sponsors-google**.

The panelists are:

- **Deniz Altınbüken** (She/Her) is a Senior Software Engineer and Tech Lead in the Storage Infrastructure team at Google, working on structured storage and employing Machine Learning to advance systems. Deniz received her PhD from Cornell University in 2017 and is an active member of the Distributed Systems community.
- **Jack Humphries** (He/Him) is a Software Engineer on the Kernel Scheduler team in Google Cloud, working on systems performance. Jack is also a first-year Ph.D. student in Computer Science at Stanford University.
- **Kathryn McKinley** (She/Her) is a Principal Research Scientist in Google Cloud, working on balancing availability, performance, and efficiency. She received her PhD, MS, and BA from Rice University and previously was an Endowed Professor at The University of Texas at Austin, a faculty member at the University of Massachusetts, Amherst, and a Principal Researcher at Microsoft.
- **Jeff Mogul** (He/Him) is a Principal Software Engineer in Google's Network Infrastructure team, working on improving and expanding our use of model-based system management. Prior to joining Google in 2013, Jeff got a PhD at Stanford and worked in corporate research at DEC and HP.
- **Hank Levy** (He/Him) is a Distinguished Software Engineer co-leading SystemsResearch@Google (SRG) from Seattle. Hank is faculty (on leave) and former Head of the Allen School for Computer Science & Engineering at University of Washington, co-founded two startups, and spent his early career at Digital Equipment Corporation.