

Kevin Klues

Education: **Ph.D, Computer Science**
[OS and Runtime Support for Efficiently Managing Cores in Parallel Applications](#)
University of California, Berkeley
August 2015

Master of Science, Computer Engineering
[Link Layer Support For Unified Radio Power Management in Wireless Sensor Networks](#)
Washington University in St. Louis
May 2007

Masters Related Research
Technische Universität Berlin
Winter 2004 - Summer 2005 (3 Semesters Completed)

Bachelor of Science, Electrical Engineering and Computer Science
Rose-Hulman Institute of Technology
Graduated Summa Cum Laude, November 2002

Languages: English – Native
German – Fluent
Spanish – Conversational

Github: <https://github.com/klueska>

Experience: **Google Inc.**
Software Engineer *June '15 - Present*

- Developer on the Akaros Team - A node OS for the datacenter
- Akaros was a primary focus of my Ph.D, and Google has allowed me to continue working on it full time
- I work on all aspects of Akaros, with a focus on developing and maintaining its low-level user-space libraries (parlib) and it's port to the Go programming language
- More info on Akaros can be found below under "Projects"

University of California, Berkeley
Graduate Student Researcher *July '08 – August '15*

- Co-design of OS and user-level library support for parallel applications on manycore platforms.. A primary focus was placed on applications in the datacenter.
- Projects: Akaros, Parlib, and Lithe (See below)

Google Inc.
Software Engineering Intern *Summer '12*

- Developer for the Systems Performance Team
- Designed and implemented a cooperative threading framework that was compatible with their existing preemptive threading framework
- Focused on improving the speed of their primary search engine, but implemented generically to be usable by all google applications

CSIRO Australia

Software Engineer

June '10 – January '11

- Developer in charge of porting TinyOS to the CSIRO's new hardware platform (opal)
- Worked on site from May '10 – January '11
- Continued to work as a consultant while back in Berkeley

WSN Lab Berkeley (Sponsored by Pirelli and Telecom Italia)

TinyOS Developer

September '08 – December '08

- Optimizing the TinyOS software stack for the SPINE Body Sensor Network project
- Integrating TOSThreads and other TinyOS 2.1 features into the SPINE framework

Stanford University

Research Assistant

July '07 – July '08

- Maintainer of TinyOS related tools and source code
- Conducted research in energy efficient OS design for wireless sensor networks
- Designed and implemented the TOSThreads library for TinyOS 2.x

Washington University in St. Louis

Sensor Network Testbed Designer

Spring '07

- Oversaw deployment of a 50+ wireless sensor network testbed based on the TWIST architecture from the Technical University of Berlin.
- Wrote initial start up scripts, set up server, got approval for placement of nodes
- Delegated responsibilities among 6 PhD and Masters students for deployment

Technische Universität Berlin

Thesis Work, Internship

January '04 – August '06

- Designed and implemented first generation radio stack for eyesIFX WSN nodes
- Collaborated with UC Berkeley to help design the second generation of the TinyOS operating system, TinyOS-2.x
- Wrote documentation and helped package software suite for the eyesIFX developers kit available from Infineon Technologies
- Designed and implemented a general purpose application for visualizing localization techniques used in WSNs

Johns Hopkins Applied Physics Lab

Engineering Intern

Summer '03

- Developed set of application tools for use by the Navy's Ship Self Defense System
- Received official letter of appreciation from the laboratory for the work completed
- Continued to work remotely while attending graduate school

Motorola

Engineering Intern

Summer '00, Summer '01

- Worked with the Motorola DSP56300, porting assembly code into C
- Designed tool for profiling load of tasks running on the DSP
- Created various other tools, wrote automated performance tests, and updated tutorials

Comp Skills: C, C++, Go, Linux / Akaros Kernel Hacking, Python, nesC, TinyOIS, JQuery/Javascript, HTML/CSS, Django, Java, Scheme, Perl

Honors: Stanford Graduate Student Fellowship (Offered, but Declined)
Rose-Hulman Presidential Scholarship
Motorola Scholarship
Washington University Distinguished Master of Science Fellowship

Affiliations: **IEEE Student Member**
ACM Student Member
Tau Beta Pi - Vice President (2001) - National Engineering Honors Fraternity
Eta Kappa Nu - National Electrical Engineering Honors Fraternity
Pi Mu Epsilon - National Mathematics Honors Fraternity
Alpha Lambda Delta - Academic Honor Society for college freshman

Projects: **Akaros** (Operating System for Manycore)
Akaros is an operating system designed for many-core architectures and large-scale SMP systems. It's primary innovation comes in its treatment of cores to provide *true* isolation for parallel applications and enable efficient user-level scheduling for more developer control. It's abstractions match particularly well to running a higher-level cluster OS (such as Mesos or Borg) on top of it.

Parlib (Efficient user-level scheduling for Akaros and Linux)

Parlib is a parallel runtime development framework for Linux and Akaros. It's primary goal is to resurrect user-level threading as a viable mechanism for exploiting parallelism in modern-day systems. Parlib-based schedulers are able to outperform the Linux NPTL across a number of representative application workloads. This includes a 3.4x faster thread context switch time, competitive performance for the NAS parallel benchmark suite, and a 6% increase in throughput over nginx for a simple thread-based webserver we wrote. These benchmarks provide substantial evidence in favor of our approach to user-level scheduling – on both Akaros and Linux, alike.

Lithe (Enabling Efficient Composition of Parallel Libraries)

Lithe is like Mesos for sharing cores *inside* an application. Applications built by composing different parallel libraries perform poorly when those libraries interfere with one another by obviously using the same physical cores. Such functionality is essential in today's open source community, where highly-tuned libraries built with competing parallel runtimes are readily available. Applications writers should be able to leverage these libraries instead of writing their own or manually tuning them to operate together efficiently. Lithe was developed to provide basic primitives and a standard interface for composing such parallel libraries efficiently. Legacy parallel libraries can be ported into the Lithe framework to provide bolt-on composability without the need to change existing application code. Ports currently exist for both Intel's Thread Building blocks (TBB) and OpenMP.

TinyOS-2.x (Operating system for Wireless Sensor Networks)

This is the successor to TinyOS 1.x, the de facto standard operating system for wireless sensor network research in academia. I was involved in this project since it began in 2004 until I left in 2010. I designed and implemented its resource and power management subsystems, its printf library, and its TOSThreads thread library.

Publications: Academic Milestones:

Kevin Klues. *OS and Runtime Support for Efficiently Managing Cores in Parallel Applications*. Ph.D Thesis. EECS Department, University of California, Berkeley. August 2015.

Kevin Klues. *Link Layer Support For Unified Radio Power Management in Wireless Sensor Networks*. Master's Thesis. Department of Computer Science and Engineering, Washington University in St. Louis. May 2007.

Journal Papers:

Raja Jurdak, **Kevin Klues**, Brano Kusy, Christian Richter, Koen Langendoen, and Michael Brünig. *Opal: A Multi-radio Platform for High Throughput Wireless Sensor Networks*. To appear in *IEEE Embedded Systems Letters*. 2011.

Kevin Klues, Guoliang Xing, and Chenyang Lu. *Link Layer Driver Architecture for Unified Radio Power Management in Wireless Sensor Networks*. ACM Transactions on Embedded Computing Systems, accepted.

Guoliang Xing, Mo Sha, Greg Hackmann, **Kevin Klues**, Octav Chipara, and Chenyang Lu. *Towards Unified Radio Power Management for Wireless Sensor Networks*. Wireless Communications and Mobile Computing (WCMC), special issue on Distributed Systems of Sensors and Actuators, 9(3):313-323, March 2009.

Conference Proceedings:

JeongGil Ko, **Kevin Klues**, Christian Richter, Wanja Hofer, Brano Kusy, Michael Brünig, Thomas Schmid, Q.Wang, Prabal Dutta, Andreas Terzis. *Low Power or High Performance? A Tradeoff Whose Time Has Come (and Nearly Gone)*. In proceedings for the European Conference on Wireless Sensor Networks. (EWSN 2012).

Barret Rhoden, **Kevin Klues**, David Zhu, and Eric Brewer. *Improving Per-Node Efficiency in the Datacenter with New OS Abstractions*. In proceedings for The 2nd ACM Symposium on Cloud Computing. (Position Paper) (SOCC 2011).

Kevin Klues, Chieh-Jan Mike Liang, Jenogyueup Paek, Razvan Musaloiu-E, Ramesh Govindan, Philip Levis, and Andreas Terzis. *TOSThreads: Safe and Non-invasive Preemption in TinyOS*. In proceedings for The 7th ACM Conference on Embedded Networked Sensor Systems (Sensys 2009).

Kevin Klues, Vlado Handziski, Chenyang Lu, Adam Wolisz, David Culler, David Gay, and Phil Levis. *Integrating Concurrency Control and Energy Management in Device Drivers*. In proceedings for The 21st ACM Symposium on Operating Systems Principles (SOSP 2007).

Kevin Klues, Greg Hackmann, Octav Chipara, and Chenyang Lu. *A Component-Based Architecture for Power-Efficient Media Access Control in Wireless Sensor Networks*. In proceedings for The 5th ACM Conference on Embedded Networked Sensor Systems (SenSys 2007).

Kevin Klues, Guoliang Xing, and Chenyang Lu. *Link Layer Support for Flexible Radio Power Management in Wireless Sensor Networks*. In proceedings for The Fifth International Conference on Information Processing in Sensor Networks (IPSN 2007).

Workshop Proceedings:

Kevin Klues, Barret Rhoden, Andrew Waterman, David Zhu, Eric Brewer. *Processes and Resource Management in a Scalable Manycore OS*. In the second USENIX Workshop on Hot Topics in Parallelism (HotPar 2010).

Rose Liu, **Kevin Klues**, Steven Hofmeyr, Sarah Bird, Krste Asanovic, and John Kubiatiowicz. *Tessellation: Space-Time Partitioning in a Manycore Client OS*. In Proceedings of the first USENIX Workshop on Hot Topics in Parallelism (HotPar 2009).

Kevin Klues, Guoliang Xing, and Chenyang Lu. *Toward a Unified Radio Power Management Architecture for Wireless Sensor Networks*. In proceedings for The First International Workshop on Wireless Sensor Network Architecture (WWSNA 2007).

Conference Posters:

Kevin Klues, Barret Rhoden, David Zhu, Paul Pearce, Eric Brewer, and John Kubiatiowicz. *Tessellation OS: Abstractions for Scalable Operating Systems on Manycore Architectures*. Presented as part of a Work in Progress (WIP) at The 22nd ACM Symposium on Operating Systems Principles (SOSP 2009).

Kevin Klues, Guoliang Xing, Chenyang Lu. *A Unified Architecture for Flexible Radio Power Management in Wireless Sensor Networks*. In Proceedings for the 4th ACM Conference on Embedded Networked Sensor Systems (Sensys 2006).

The TinyOS Alliance (presented by **Kevin Klues**). *TinyOS 2.1: Adding Threads and Memory Protection to TinyOS*. In Proceedings for the 6th ACM Conference on Embedded Networked Sensor Systems (Sensys 2008).

Kevin Klues. *Power Locks: Enabling Ultra Low Power Embedded Systems Applications with Minimal Developer Effort*. Stanford CS Forum. March 2008.

Technical Reports:

Kevin Klues, Vlado Handziski, David Culler, David Gay, Phil Levis, Chenyang Lu, Adam Wolisz. *Dynamic Resource Management in a Static Network Operating System*. Technical Report WUCSE-2006-56, Washington University in St. Louis, Oct. 2006.

P. Levis, D. Gay, V. Handziski, J.-H.Hauer, B.Greenstein, M.Turon, J.Hui, **K.Klues**, C.Sharp, R.Szewczyk, J.Polastre, P.Buonadonna, L.Nachman, G.Tolle, D.Culler, and A.Wolisz. *T2: A Second Generation OS For Embedded Sensor Networks*. Technical Report TKN-05-007, Telecommunication Networks Group, Technische Universität Berlin, Nov. 2005

Other Works: TinyOS Tutorials:

The TOSThreads Thread Library
Resource Arbitration and Power Management
The TinyOS Printf Library
Writing Low-Power Applications

TinyOS Enhancement Proposals:

TEP105: Low Power Listening
TEP108: Resource Arbitration
TEP115: Power Management of Non-Virtualized Devices
TEP128: Platform Independent Non-Volatile Storage Abstractions
TEP134: The TOSThreads Thread Library

Noted Talks: TOSThreads: Thread-safe and Non-invasive Preemption in TinyOS

- The 7th ACM Conference on Embedded Networked Sensor Systems (SenSys 2009)
- CSIRO Australia (July 2010)

Abstractions for Scalable OSs on Manycore Architectures

- WIP at The 22nd ACM Symposium on Operating Systems Principles (SOSP 2009)

TinyOS 2.1 Tutorial (TOSThreads)

- The 8th ACM / IEEE International Conference on Information Processing in Sensor Networks (April 2009)

The TinyOS Documentation Working Group

- TinyOS Technology Exchange (TTX) V (February 2008)

TinyOS 2.x Contrib

- TinyOS Technology Exchange (TTX) V (February 2008)
- The TinyOS Operating System: Past, Present, and Future Directions
- Universität Erlangen-Nürnberg, Erlangen Germany (February 2008)
- Universität Karlsruhe, Karlsruhe Germany (February 2008)
- Technische Universität Berlin, Berlin Germany (February 2008)

A Component-Based Architecture for Power-Efficient Media Access Control in WSNs

- The 5th ACM Conference on Embedded Networked Sensor Systems (SenSys 2007)

Integrating Concurrency Control and Energy Management in Device Drivers.

- Universität Erlangen-Nürnberg, Erlangen Germany (February 2008)
- Universität Karlsruhe, Karlsruhe Germany (February 2008)
- The 21st ACM Symposium on Operating Systems Principles (SOSP 2007).

Link Layer Support For Unified Radio Power Management in Wireless Sensor Networks.

- The 5th Int'l Conference on Information Processing in Sensor Networks (IPSN 2007)

Towards a Unified Radio Power Management Architecture for Wireless Sensor Networks.

- The 1st Int'l Workshop on Wireless Sensor Network Architecture (WWSNA 2007)

Teaching: Operating Systems TA

- Lead TA in the CS162 Operating Systems course at UC Berkeley, Fall 2013

Hands-On TinyOS

- A three day mini-course on using TinyOS and all of its advanced features
- Lectures included an introduction to TinyOS, TOSThreads, and the integrated concurrency control and energy management (ICEM) features of TinyOS
- Run as a guest lecture course at RWTH Aachen (3/Feb/2009 – 5/Feb/2009)

References: **Eric Brewer**

Ph.D advisor

417 Sutardja Dai Hall, UC Berkeley

Berkeley, CA 94720-1758

United States

Email: brewer@cs.berkeley.edu

Homepage: <http://www.cs.berkeley.edu/~brewer/>

Krste Asanovic

Ph.D Qualifying Exam Chair

579 Soda Hall, MC #1776

Berkeley, CA 94720-1758

United States

Email: krste@cs.berkeley.edu

Homepage: <http://www.cs.berkeley.edu/~krste/>

Phil Levis

Former Supervisor and Leader of the TinyOS project

358 Gates Hall, Stanford University

Stanford, CA, 94305-9030

United States

Email: pal@cs.stanford.edu

Homepage: <http://csl.stanford.edu/~pal/>

MORE REFERENCES GIVEN UPON REQUEST