7.3 Zachary Tatlock

Zachary Tatlock has not received prior NSF funding.

Project personnel

1. Zachary Tatlock; University of Washington; PI
2. Thomas E. Anderson; University of Washington; co-PI
3. Michael D. Ernst; University of Washington; co-PI
4. Suzanne Millstein; University of Washington; Other professional (staff programmer)
5. Calvin Loncaric; University of Washington; Graduate student (research assistant)
6. Chandrakana Nandi; University of Washington; Graduate student (research assistant)
7. Pavel Panchekha; University of Washington; Graduate student (research assistant)
8. Stuart Pernsteiner; University of Washington; Graduate student (research assistant)
9. Konstantin Weitz; University of Washington; Graduate student (research assistant)
10. James R. Wilcox; University of Washington; Graduate student (research assistant)
11. Doug Woos; University of Washington; Graduate student (research assistant)
12. Steve Anton; University of Washington; Undergraduate student
13. Alex Sanchez-Stern; University of Washington; Undergraduate student
14. Daryl Zuniga; University of Washington; Undergraduate student
15. Juliet Oh; International High School; High school student

Collaborators

1. Collaborators for Zachary Tatlock; University of Washington; PI

    1. Sonya Alexandrova; University of Washington
    2. Thomas Anderson; University of Washington
    3. Maya Cakmak; University of Washington
    4. Adam Chlipala; Massachusetts Institute of Technology
    5. Michael D. Ernst; University of Washington
    6. Dan Grossman; University of Washington
    7. Jon Jacky; University of Washington
    8. Dongseok Jang; University of California, San Diego
    9. David Lazar; Massachusetts Institute of Technology
    10. Sorin Lerner; University of California, San Diego
    11. Calvin Loncaric; University of Washington
    12. Eric Mullen; University of Washington
    13. Pavel Panchekha; University of Washington
    14. Stuart Pernsteiner; University of Washington
    15. Daniel Ricketts; University of California, San Diego
    16. Valentin Robert; University of California, San Diego
    17. Alex Sanchez-Stern; University of Washington
    18. Michael Stepp; University of California, San Diego
    19. Ross Tate; Cornell University
    20. Emina Torlak; University of Washington
    21. Xi Wang; University of Washington
    22. James R. Wilcox; University of Washington
    23. Doug Woos; University of Washington
    24. Nickolai Zeldovich; Massachusetts Institute of Technology

2. Collaborators for Thomas E. Anderson; University of Washington; co-PI

    1. Ivan Beschastnikh; University of British Columbia
    2. Ken Birman; Cornell
    3. Matthew Caesar; University of Illinois Urbana-Champaign
    4. Justin Cappos; NYU-Poly
    5. Dave Choffnes; Northeastern
    6. Doug Comer; Purdue
    7. Chase Cotton; Delaware
    8. Mike Dahlin; Google
    9. Colin Dixon; Brocade Networks
    10. Lucas Dixon; Google
    11. Mike Ernst; University of Washington
    12. Nick Feamster; Georgia Tech
    13. Mike Freedman; Princeton
    14. Andreas Haeberlen; University of Pennsylvania
    15. Daniel Halperin; University of Washington
    16. Zack Ives; University of Pennsylvania

17. Ethan Katz-Bassett; University of Southern California
18. Arvind Krishnamurthy; University of Washington
19. Taesoo Kim; Georgia Tech
20. William Lehr; MIT
21. Boon Thau Loo; University of Pennsylvania
22. Harsha Madhyastha; University of Michigan
23. David Mazieres; Stanford
24. Antonio Nicolosi; Stevens Institute of Technology
25. Simon Peter; University of Texas-Austin
26. Dan Ports; University of Washington
27. Sylvia Ratnasamy; University of California at Berkeley
28. Timothy Roscoe; ETH Zurich
29. Jonathan Smith; University of Pennsylvania
30. Ion Stoica; University of California Berkeley
31. Zach Tatlock; University of Washington
32. Robbert van Renesse; Cornell
33. Michael Walfish; NYU
34. Xi Wang; University of Washington
35. Hakim Weatherspoon; Cornell
36. David Wetherall; Google
37. Christopher Yoo; University of Pennsylvania

3. Collaborators for Michael D. Ernst; University of Washington; co-PI

1. Jenny Abrahamson; Microsoft
2. Ruth Anderson; University of Washington
3. Thomas E. Anderson; University of Washington
4. Shay Artzi; Zappix
5. Vinay Augustine; ABB
6. Richard Bailey; University of Washington
7. Magdalena Balazinska; University of Washington
8. Paulo Barros; Federal University of Pernambuco
9. Ivan Beschastnikh; University of British Columbia
10. Ravi Bhoraskar; Facebook
11. Yuriy Brun; University of Massachusetts
12. Yingyi Bu; Couchbase
13. Brian Burg; Apple
14. Juan Caballero; IMDEA Software Institute
15. Aaron Cammarata; VoidALPHA
16. John Cheng; Veracient LLC
17. Raymond Cheng; University of Washington
18. Seth Cooper; Northeastern University
19. Forrest Coward; Microsoft
20. Marcelo d'Amorim; Federal University of Pernambuco
21. Drew Dean; SRI International
22. Werner Dietl; University of Waterloo

23. Stephanie Dietzel; Tableau Software
24. Kellen Donohue; Google
25. Leonard Eusebi; Charles River Analytics
26. Gordon Fraser; University of Sheffield
27. Vijay Ganesh; University of Waterloo
28. Colin S. Gordon; Drexel University
29. Dan Grossman; University of Washington
30. Sean Guarino; Charles River Analytics
31. Philip J. Guo; University of Rochester
32. Seungyeop Han; University of Washington
33. Irfan Ul Haq; IMDEA Software Institute
34. Pingyang He; University of Washington
35. Reid Holmes; University of British Columbia
36. Pieter Hooimeijer; University of Virginia
37. Bill Howe; University of Washington
38. Wei Huang; Google
39. Laura Inozemtseva; University of Waterloo
40. Jon Jacky; University of Washington
41. Darioush Jalali; University of Washington
42. Ralph E. Johnson; University of Illinois
43. René Just; University of Massachusetts
44. Andrew Keplinger; Left Brain Games
45. Adam Kieżun; Broad Institute
46. Gene Kim; University of Washington
47. Andrew J. Ko; University of Washington
48. Karl Koscher; University of California at San Diego
49. Arvind Krishnamurthy; University of Washington
50. Wing Lam; University of Illinois
51. Dominic Langenegger; ETH Zurich
52. Jingyue Li; DNV Research & Innovation
53. Nuo Li; ABB
54. Calvin Loncaric; University of Washington
55. Alberto Lovato; Università di Verona
56. Hao Lü; Google
57. Damiano Macedonio; Julia Srl
58. Thomas Maddern; Veracient LLC
59. Kelly McLaughlin; XPD Analytics
60. Massimo Merro; Università degli Studi di Verona
61. Ana Milanova; Rensselaer Polytechnic Institute
62. Suzanne Millstein; University of Washington
63. Nathaniel Mote; University of Washington
64. John Murray; SRI International
65. Kıvanç Muşlu; Microsoft
66. David Notkin; deceased
67. Robert Ordóñez; Southern Adventist University
68. Johan Östlund; Uppsala University

69. Pavel Panchekha; University of Washington
70. Timothy Pavlik; University of Washington
71. Jeff H. Perkins; MIT
72. Stuart Pernsteiner; University of Washington
73. Paul Pham; The Evergreen State College
74. Amarin Phaosawasdi; University of Illinois
75. Zoran Popović; University of Washington
76. Alex Potanin; Victoria University of Wellington
77. Brian Robinson; ABB
78. Franziska Roesner; University of Washington
79. Todd W. Schiller; Bridgewater Associates
80. Sigurd Schneider; Saarland University
81. Will Scott; University of Washington
82. Michael Sloan; University of Washington
83. Ciprian Spiridon; Julia Srl
84. Eric Spishak; Google
85. Fausto Spoto; Università di Verona
86. Siwakorn Srisakaokul; University of Illinois
87. Luke Swart; University of Washington
88. Zachary Tatlock; University of Washington
89. Javier Thaine; University of Washington
90. Emina Torlak; University of Washington
91. Ben Treibelhorn; Seattle University
92. Mohsen Vakilian; Google
93. Paul Vines; University of Washington
94. Brian Walker; University of Washington
95. Xi Wang; University of Washington
96. Ronald Watro; Raytheon BBN
97. Konstantin Weitz; University of Washington
98. James R. Wilcox; University of Washington
99. Steven A. Wolfman; University of British Columbia
100. Doug Woos; University of Washington
101. Edward X. Wu; Extrahop
102. Jochen Wuttke; Google
103. Cheng Zhang; Shanghai Jiao Tong University
104. Sai Zhang; Google
105. Yoav Zibin; Google

Collaboration plan

The PIs are Zachary Tatlock, Michael Ernst, and Tom Anderson. We have an established, productive history of collaboration and co-authored the PLDI 2015 paper [WWP+15a] that inspired this grant proposal. We all work in the same department, with offices within a few feet of each other. We co-advise students and work together on multiple joint projects across a number of distinct areas of computer science, including verification, software engineering, and distributed systems. We participate in multiple joint weekly meetings, both on the Verdi project and on other current research topics. In particular, Tatlock and Ernst co-advise students Pavel Panchekha and Konstantin Weitz. Anderson and Ernst co-advise student Doug Woos, with Tatlock as an informal advisor. Tatlock is an informal advisor to Ernst's student Calvin Loncaric, and Ernst is an informal advisor to Tatlock's students Stuart Pernsteiner and James Wilcox. All of these students are interested in formal verification of distributed systems and will work on the proposed project. Ernst and Anderson also co-advised Ivan Beschastnikh (who graduated in 2013 and is currently a professor at UBC), whose work spans distributed systems and software engineering.

The PIs will jointly define and participate in the graduate seminars on systems, programming languages, and software engineering, which are held each quarter. In Spring 2015, Zach Tatlock, along with our colleague Xi Wang, will develop and teach a graduate seminar, CSE 599, on systems verification focusing on operating systems, distributed systems, and networking. This seminar will be project-based, and students will be encouraged to use Verdi to verify their own systems or work on further developing a part of the Verdi framework.

The Software Engineering and Programming Languages Research Lab is a place where students "hang out," allowing not only in-depth interaction among the team members but also informal interactions and idea-sharing with other students in these areas. The PIs bring different styles, knowledge, and approach to the research table. This allows us to help students develop along multiple dimensions. We anticipate many technical challenges in this ambitious project — but luckily collaboration won't be an obstacle.

7.4 Division of Work

As described in the proposal, we will investigate 10 research questions. The primary faculty leads for each question are listed below, but we stress that everyone on the team is a generalist. Although we each have different areas of expertise, we all intend to pitch in on whatever sub-project is needed to push the goals of the work forward. We strive to foster the same attitude among our graduate students.

RQ1	(Anderson, Tatlock)	**RQ6**	(Ernst, Tatlock)
RQ2	(Anderson, Tatlock)	**RQ7**	(Anderson, Tatlock)
RQ3	(Anderson, Ernst)	**RQ8**	(Ernst, Tatlock)
RQ4	(Ernst, Tatlock)	**RQ9**	(Ernst, Tatlock)
RQ5	(Anderson, Ernst)	**RQ10**	(Anderson, Ernst)

Each of these questions has a multi-year research and development agenda. Instead of trying to complete one area and then the next, our plan is to work on them in parallel as the results from many of these questions can help developing solutions to the rest. We will split the research agenda in each area into a series of six-month challenges that we can integrate into the mainline Verdi framework if successful, and learn from our failures if not. The idea is to repeatedly course-correct, to go from working system to working system.

7.5 Collaboration Mechanisms

We plan to continue the model we have used in the recent past:

- Graduate students will be co-advised in almost all cases. Not only do students get the benefit of multiple perspectives, but this provides an almost daily opportunity for faculty to discuss issues with respect to the project.
- Graduate researchers will typically work on sub-projects in small teams of 2–3; in our view, student collaboration across project topics is as advantageous as faculty collaboration.
- Undergraduate researchers will be paired with a graduate student mentor, overseen by one other faculty member. We have found that undergraduates benefit from more frequent interaction than is often possible with over-committed faculty. Furthermore, graduate students learn a lot from the experience of being an advisor.
- We will have weekly project meetings, including faculty and students, to discuss research progress and plan future work. We believe students should see the process of how decisions are made, rather than making those decisions behind closed doors.
- We will hold weekly PI meetings to track progress against progress goals, diagnose and fix student issues, and strategize industrial adoption.
- We will organize and hold an annual off-site meeting to coordinate with practioners from our industrial partners.

Data management plan

The goal of this project is to enable the creation of formally-proven distributed system implementations. To evaluate the space of design alternatives, we plan to build proofs and extract runnable source code from them; we will also run experiments on the running systems. All source code, along with configuration files and instructions for reproducing published measurements of the system, will be made public, hosted via GitHub and accessible from the project website at the University of Washington.

The principal investigators have a long history of making research software and data publicly available, so that other scientists can reproduce and build on the work. Anderson has led the development of several widely used open source software artifacts over the past few years, most notably BitTyrant [PIA+07] and OneSwarm [IPKA10]. Both have been downloaded more than a hundred thousand times. Most recently they have led an effort to produce an easy to configure public anti-censorship tool, uProxy, to enable client browsers to work together to evade censorship [Upr]. uProxy is open source software, and it has more than ten thousand enrolled beta users for an initial release in the next few months. They have also led the iPlane Internet topology mapping project [MIP+06]. iPlane has collected and published Internet topology maps continuously since 2006. The iPlane system measures millions of paths per day and the results have been used by more than thirty research groups. Ernst recently led the development of the Checker Framework (http://checkerframework.org/), which is downloaded about 300 times per day, and his Daikon (http://plse.cs.washington.edu/daikon/) and Randoop (http://mernst.github.io/randoop/) projects are even more popular. Dozens more software downloads are listed at http://homes.cs.washington.edu/~mernst/software/ and http://homes.cs.washington.edu/~mernst/pubs/.

As NSDI steering committee chair, Anderson was instrumental in encouraging NSDI to create a special award for those papers whose source code and data sets are placed in the public domain. Tatlock is the Artifact Evaluation Committee Co-Char for PLDI 2016. In this role, he will encourage the authors of all accepted papers to submit their implementations to the committee which will attempt to independently replicate the paper's results before the conference is held.

Policies for access and sharing. We will make the source code for our tools and all test configurations available for public download using the BSD or Apache license. To ensure the broadest impact of our work, we will make the source code available for re-use without restriction.

Reproducibility of experimental results. We will publish our results and algorithms using peer-reviewed venues such as PLDI, POPL, ICSE, FSE, OSDI, and NSDI, as well as through technical reports freely available through our department web sites. All published experimental results will be accompanied by public source code, configuration files, and data sets, allowing others to directly reproduce our results.

Plans for archiving data. Data archival is essential for the repeatability of experiments and longitudinal studies. We will ensure archival of the measured data reported in our published work. We expect these data sets to be small enough to be served out of the same GitHub and department repositories for the project source code.

Standards, data format and content. There is no accepted standard for storing and maintaining proofs (other than the source code of the proof in Coq) and experimental data from distributed systems. We will ensure that the data set and source code distribution includes sufficient documentation to allow for the published data to be reconstructed by a third party.

References

[ABF+05] Brian E. Aydemir, Aaron Bohannon, Matthew Fairbairn, J. Nathan Foster, Benjamin C. Pierce, Peter Sewell, Dimitrios Vytiniotis, Geoffrey Washburn, Stephanie Weirich, and Steve Zdancewic. Mechanized metatheory for the masses: The poplmark challenge. In *Proceedings of the 18th International Conference on Theorem Proving in Higher Order Logics*, TPHOLs'05, pages 50–65, Berlin, Heidelberg, 2005. Springer-Verlag.

[ADH+14] Andrew W. Appel, Robert Dockins, Aquinas Hobor, Lennart Beringer, Josiah Dodds, Gordon Stewart, Sandrine Blazy, and Xavier Leroy. *Program Logics for Certified Compilers*. Cambridge University Press, New York, NY, USA, 2014.

[Ama11] Amazon. Summary of the Amazon EC2 and Amazon RDS service disruption in the US East Region, April 2011. http://aws.amazon.com/message/65648/.

[BBE+11a] Ivan Beschastnikh, Yuriy Brun, Michael D. Ernst, Arvind Krishnamurthy, and Thomas E. Anderson. Bandsaw: Log-powered test scenario generation for distributed systems. In *SOSP Work In Progress*, Cascais, Portugal, October 24–26, 2011.

[BBE+11b] Ivan Beschastnikh, Yuriy Brun, Michael D. Ernst, Arvind Krishnamurthy, and Thomas E. Anderson. Mining temporal invariants from partially ordered logs. *SIGOPS Operating Systems Review*, 45(3):39–46, December 2011.

[BBEK14] Ivan Beschastnikh, Yuriy Brun, Michael D. Ernst, and Arvind Krishnamurthy. Inferring models of concurrent systems from logs of their behavior with CSight. In *ICSE'14, Proceedings of the 36th International Conference on Software Engineering*, pages 468–479, Hyderabad, India, June 4–6, 2014.

[BBS+11] Ivan Beschastnikh, Yuriy Brun, Sigurd Schneider, Michael Sloan, and Michael D. Ernst. Leveraging existing instrumentation to automatically infer invariant-constrained models. In *ESEC/FSE 2011: The 8th joint meeting of the European Software Engineering Conference (ESEC) and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)*, pages 267–277, Szeged, Hungary, September 7–9, 2011.

[BFN+06] Steven Bishop, Matthew Fairbairn, Michael Norrish, Peter Sewell, Michael Smith, and Keith Wansbrough. Engineering with logic: HOL specification and symbolic-evaluation testing for TCP implementations. In *Proceedings of the 33rd ACM Symposium on Principles of Programming Languages (POPL)*, pages 55–66, Charleston, SC, January 2006.

[BHEN11a] Yuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin. Crystal: Precise and unobtrusive conflict warnings. In *ESEC/FSE 2011: The 8th joint meeting of the European Software Engineering Conference (ESEC) and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)*, pages 267–277, Szeged, Hungary, September 7–9, 2011.

[BHEN11b] Yuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin. Proactive detection of collaboration conflicts. In *ESEC/FSE 2011: The 8th joint meeting of the European Software Engineering Conference (ESEC) and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)*, pages 168–178, Szeged, Hungary, September 7–9, 2011.

[BHEN13] Yuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin. Early detection of collaboration conflicts and risks. *IEEE Transactions on Software Engineering*, 39(10):1358–1375, October 2013.

[Blo83] Toby Bloom. *Dynamic Module Replacement in a Distributed Programming System*. Ph.D., MIT, 1983. Also available as MIT LCS Tech. Report 303.

[BMH+12] Yuriy Brun, Kıvanç Muşlu, Reid Holmes, Michael D. Ernst, and David Notkin. Predicting development trajectories to prevent collaboration conflicts. In *The Future of Collaborative Software Development*, Bellevue, WA, USA, February 12, 2012.

[Bro08] Martin Brown. Pakistan hijacks YouTube. http://research.dyn.com/2008/02/pakistan-hijacks-youtube-1/, 2008. Accessed: 2015-02-20.

[Bur06] Mike Burrows. The Chubby lock service for loosely-coupled distributed systems. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI)*, Seattle, WA, November 2006.

[CAB+86] R. L. Constable, S. F. Allen, H. M. Bromley, W. R. Cleaveland, J. F Cremer, R. W. Harper, D. J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, J. T Sasaki, and S. F. Smith. *Implementing Mathematics with The Nuprl Proof Development System*. Prentice Hall, 1986.

[CDG+08] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.*, 26(2):4:1–4:26, June 2008.

[CGR07] Tushar D. Chandra, Robert Griesemer, and Joshua Redstone. Paxos made live: An engineering perspective. In *Proceedings of the 26th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, pages 398–407, Portland, OR, August 2007.

[Chl11] Adam Chlipala. Mostly-automated verification of low-level programs in computational separation logic. In *Proceedings of the 2011 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pages 234–245, San Jose, CA, June 2011.

[Chl13] Adam Chlipala. *Certified Programming with Dependent Types*. MIT Press, December 2013.

[Cor14] etcd: A highly-available key value store for shared configuration and service discovery, 2014. https://github.com/coreos/etcd.

[Cow10] Jim Cowie. China's 18-minute mystery. http://research.dyn.com/2010/11/chinas-18-minute-mystery/, 2010. Accessed: 2015-03-21.

[CS09] Bernadette Charron-Bost and André Schiper. The heard-of model: computing in distributed systems with benign faults. *Distributed Computing*, 22(1):49–71, 2009.

[DDE+11] Werner Dietl, Stephanie Dietzel, Michael D. Ernst, Kıvanç Muşlu, and Todd Schiller. Building and using pluggable type-checkers. In *ICSE'11, Proceedings of the 33rd International Conference on Software Engineering*, pages 681–690, Waikiki, Hawaii, USA, May 25–27, 2011.

[DHJ+] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo: Amazon's highly available key-value store. SOSP '07.

[DLE03] Nii Dodoo, Lee Lin, and Michael D. Ernst. Selecting, refining, and evaluating predicates for program analysis. Technical Report MIT-LCS-TR-914, MIT Laboratory for Computer Science, Cambridge, MA, July 21, 2003.

[ECGN01] Michael D. Ernst, Jake Cockrell, William G. Griswold, and David Notkin. Dynamically discovering likely program invariants to support program evolution. *IEEE Transactions on Software Engineering*, 27(2):99–123, February 2001. A previous version appeared in *ICSE '99, Proceedings of the 21st International Conference on Software Engineering*, pages 213–224, Los Angeles, CA, USA, May 19–21, 1999.

[EPG+07] Michael D. Ernst, Jeff H. Perkins, Philip J. Guo, Stephen McCamant, Carlos Pacheco, Matthew S. Tschantz, and Chen Xiao. The Daikon system for dynamic detection of likely invariants. *Science of Computer Programming*, 69(1–3):35–45, December 2007.

[FGL12] S.R. Foster, William G. Griswold, and S. Lerner. Witchdoctor: Ide support for real-time auto-completion of refactorings. In *Software Engineering (ICSE), 2012 34th International Conference on*, June 2012.

[GAM+07] Dennis Geels, Gautam Altekar, Petros Maniatis, Timothy Roscoe, and Ion Stoica. Friday: Global comprehension for distributed replay. In *Proceedings of the 4th Symposium on Networked Systems Design and Implementation (NSDI)*, pages 285–298, Cambridge, MA, April 2007.

[GEG13] Colin S. Gordon, Michael D. Ernst, and Dan Grossman. Rely-guarantee references for refinement types over aliased mutable data. In *Proceedings of the 34th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '13, pages 73–84, New York, NY, USA, 2013. ACM.

[GL00] Stephen J. Garland and Nancy Lynch. Using I/O automata for developing distributed systems. In *Foundations of Component-based Systems*. Cambridge University Press, 2000.

[GMT09] G. Gonthier, A. Mahboubi, , and E. Tassi. A Small Scale Reflection Extension for the Coq System. Technical Report 645, Microsoft Research - Inria Joint Centre, 2009.

[Gol14] Sharon Goldberg. Why is it taking so long to secure Internet routing? *Queue*, 12(8):20:20–20:33, August 2014.

[GRF13] Arjun Guha, Mark Reitblatt, and Nate Foster. Machine-verified network controllers. In *Proceedings of the 2013 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pages 483–494, Seattle, WA, June 2013.

[Hay98] Mark Hayden. *The Ensemble System*. PhD thesis, Cornell University, 1998.

[Hig11] High Scalability. The updated big list of articles on the Amazon outage, May 2011. http://highscalability.com/blog/2011/5/2/the-updated-big-list-of-articles-on-the-amazon-outage.html.

[HLP+13] Seungyeop Han, Vincent Liu, Qifan Pu, Simon Peter, Thomas E. Anderson, Arvind Krishnamurthy, and David Wetherall. Expressive privacy control with pseudonyms. In Dah Ming Chiu, Jia Wang, Paul Barford, and Srinivasan Seshan, editors, *SIGCOMM 2013: Proceedings of the ACM SIGCOMM 2013 Conference*, pages 291–302, Hong Kong, China, August 12-16 2013. ACM.

[HLvR09] Jason J. Hickey, Nancy Lynch, and Robbert van Renesse. Specifications and proofs for Ensemble layers. In *Proceedings of the 15th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, pages 119–133, York, UK, March 2009.

[HSS+14] Christopher M. Hayden, Karla Saur, Edward K. Smith, Michael Hicks, and Jeffrey S. Foster. Kitsune: Efficient, general-purpose dynamic software updating for c. *ACM Trans. Program. Lang. Syst.*, 36(4), October 2014.

[Int] Internet2 configurations. http://vn.grnoc.iu.edu/Internet2/configs/configs.html.

[IPKA10] Tomas Isdal, Michael Piatek, Arvind Krishnamurthy, and Thomas E. Anderson. Privacy-preserving P2P data sharing with OneSwarm. In Shivkumar Kalyanaraman, Venkata N. Padmanabhan, K. K. Ramakrishnan, Rajeev Shorey, and Geoffrey M. Voelker, editors, *SIGCOMM 2010: Proceedings of the ACM SIGCOMM 2010 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 111–122. ACM, August 30 - September 3, 2010.

[Jac12] Daniel Jackson. *Software Abstractions: Logic, Language, and Analysis*. MIT Press, February 2012.

[JTL12] Dongseok Jang, Zachary Tatlock, and Sorin Lerner. Establishing browser security guarantees through formal shim verification. In *Proceedings of the 21st Usenix Security Symposium*, pages 113–128, Bellevue, WA, August 2012.

[KAB+07] Charles Killian, James W. Anderson, Ryan Braud, Ranjit Jhala, and Amin Vahdat. Mace: Language support for building distributed systems. In *Proceedings of the 2007 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pages 179–188, San Diego, CA, June 2007.

[KAJV07] Charles Killian, James W. Anderson, Ranjit Jhala, and Amin Vahdat. Life, death, and the critical transition: Finding liveness bugs in systems code. In *Proceedings of the 4th Symposium on Networked Systems Design and Implementation (NSDI)*, pages 243–256, Cambridge, MA, April 2007.

[KEH+09] Gerwin Klein, Kevin Elphinstone, Gernot Heiser, June Andronick, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Michael Norrish, Rafal Kolanski, Thomas Sewell, Harvey Tuch, and Simon Winwood. seL4: Formal verification of an OS kernel. In *Proceedings of the 22nd ACM Symposium on Operating Systems Principles (SOSP)*, pages 207–220, Big Sky, MT, October 2009.

[Kin13] Kyle Kingsbury. Call me maybe, May 2013. https://aphyr.com/posts/281-call-me-maybe-carly-rae-jepsen-and-the-perils-of-network-partitions.

[Lam98] Leslie Lamport. The part-time parliament. *ACM Trans. Comput. Syst.*, 16(2):133–169, May 1998.

[Lam02] Leslie Lamport. *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley Professional, July 2002.

[Lam14] Leslie Lamport. Thinking for programmers, April 2014. http://channel9.msdn.com/Events/Build/2014/3-642.

[LAS14] Vu Le, Mehrdad Afshari, and Zhendong Su. Compiler validation via equivalence modulo inputs. In *Proceedings of the 2014 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pages 216–226, Edinburgh, UK, June 2014.

[Ler09a] Xavier Leroy. Formal verification of a realistic compiler. *Communications of the ACM*, 52(7):107–115, July 2009.

[Ler09b] Xavier Leroy. A formally verified compiler back-end. *J. Automated Reasoning*, 43(4):363–446, December 2009.

[LFKA] Wyatt Lloyd, Michael J. Freedman, Michael Kaminsky, and David G. Andersen. Don't settle for eventual: Scalable causal consistency for wide-area storage with COPS. SOSP '11.

[LG02] N Lynch and S Gilbert. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 33:51–59, 2002.

[LGW$^+$08] Xuezheng Liu, Zhenyu Guo, Xi Wang, Feibo Chen, Xiaochen Lian, Jian Tang, Ming Wu, M. Frans Kaashoek, and Zheng Zhang. D^{3}S: Debugging deployed distributed systems. In *Proceedings of the 5th Symposium on Networked Systems Design and Implementation (NSDI)*, pages 423–437, San Francisco, CA, April 2008.

[LHKA11] Vincent Liu, Seungyeop Han, Arvind Krishnamurthy, and Thomas E. Anderson. Tor instead of IP. In Hari Balakrishnan, Dina Katabi, Aditya Akella, and Ion Stoica, editors, *HotNets 2011: Proceedings of the Tenth ACM Workshop on Hot Topics in Networks*, page 14, Cambridge, MA, USA, November 14 - 15, 2011. ACM.

[LHKA13] Vincent Liu, Daniel Halperin, Arvind Krishnamurthy, and Thomas E. Anderson. F10: A fault-tolerant engineered network. In Nick Feamster and Jeffrey C. Mogul, editors, *NSDI 2013: Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation*, pages 399–412, Lombard, IL, USA, April 2-5, 2013. USENIX Association.

[LKvR$^+$99] Xiaoming Liu, Christoph Kreitz, Robbert van Renesse, Jason Hickey, Mark Hayden, Kenneth Birman, and Robert L. Constable. Building reliable, high-performance communication systems from components. In *Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP)*, pages 80–92, Kiawah Island, SC, December 1999.

[LNZ15] David Lo, Nachiappan Nagappan, and Thomas Zimmermann. How practitioners perceive the relevance of software engineering research. In *ESEC/FSE 2015: The 10th joint meeting of the European Software Engineering Conference (ESEC) and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)*, Bergamo, Italy, September 2–4, 2015.

[LPB15] Caroline Lemieux, Dennis Park, and Ivan Beschastnikh. General LTL specification mining. In *ASE 2015: Proceedings of the 30th Annual International Conference on Automated Software Engineering*, Lincoln, NE, USA, November 11–13, 2015.

[Lyn96] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

[Mad14] Doug Madory. Chinese routing errors redirect Russian traffic. http://research.dyn.com/2014/11/chinese-routing-errors-redirect-russian-traffic/, 2014. Accessed: 2015-02-20.

[MBEN13] Kıvanç Muşlu, Yuriy Brun, Michael D. Ernst, and David Notkin. Making offline analyses continuous. In *ESEC/FSE 2013: The 9th joint meeting of the European Software Engineering Conference (ESEC) and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)*, pages 323–333, St. Petersburg, Russia, August 21–23, 2013.

[MBEN15] Kıvanç Muşlu, Yuriy Brun, Michael D. Ernst, and David Notkin. Reducing feedback delay of software development tools via continuous analysis. *IEEE Transactions on Software Engineering*, 2015.

[MBH$^+$12a] Kıvanç Muşlu, Yuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin. Improving IDE recommendations by considering global implications of existing recommendations. In *ICSE'12 New Ideas and Emerging Results Track*, pages 1349–1352, Zürich, Switzerland, June 6–8, 2012.

[MBH$^+$12b] Kıvanç Muşlu, Yuriy Brun, Reid Holmes, Michael D. Ernst, and David Notkin. Speculative analysis of integrated development environment recommendations. In *Object-Oriented Programming Systems, Languages, and Applications (OOPSLA 2012)*, pages 669–682, Tucson, AZ, USA, October 23–25, 2012.

[MBM13] Kıvanç Muşlu, Yuriy Brun, and Alexandra Meliou. Data debugging with continuous testing. In *ESEC/FSE 2013: The 9th joint meeting of the European Software Engineering Conference (ESEC) and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE)*, pages 631–634, St. Petersburg, Russia, August 21–23, 2013.

[MBM15] Kıvanç Muşlu, Yuriy Brun, and Alexandra Meliou. Preventing data errors with continuous testing. In *ISSTA 2015, Proceedings of the 2015 International Symposium on Software Testing and Analysis*, pages 373–384, Baltimore, MD, USA, July 15–17, 2015.

[MBNC14] Kıvanç Muşlu, Christian Bird, Nachi Nagappan, and Jacek Czerwonka. Transition from centralized to distributed version control systems: A case study on reasons, barriers, and outcomes. In *ICSE'14, Proceedings of the 36th International Conference on Software Engineering*, pages 334–344, Hyderabad, India, June 4–6, 2014.

[MCB14] Gregory Malecha, Adam Chlipala, and Thomas Braibant. Compositional computational reflection. In *Proceedings of the 5th International Conference on Interactive Theorem Proving*, pages 374–389, Vienna, Austria, July 2014.

[McC10] Dugald McConnell. Chinese company 'hijacked' U.S. web traffic. http://www.cnn.com/2010/US/11/17/websites.chinese.servers/, 2010. Accessed: 2015-03-21.

[MHvF15] Jedidiah McClurg, Hossein Hojjat, Pavol Černý, and Nate Foster. Efficient synthesis of network updates. In *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2015, pages 196–207, New York, NY, USA, 2015. ACM.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[MIP+06] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas E. Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iplane: An information plane for distributed services. In Brian N. Bershad and Jeffrey C. Mogul, editors, *OSDI 2006: Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, pages 367–380, Seattle, WA, USA, November 2006. USENIX Association.

[MSBE15] Kıvanç Muşlu, Luke Swart, Yuriy Brun, and Michael D. Ernst. Simplifying development history information retrieval via multi-grained views. In *ASE 2015: Proceedings of the 30th Annual International Conference on Automated Software Engineering*, Lincoln, NE, USA, November 11–13, 2015.

[MSW11] Kıvanç Muşlu, Bilge Soran, and Jochen Wuttke. Finding bugs by isolating unit tests. In *ESEC/FSE 2011: The 8th joint meeting of the European Software Engineering Conference (ESEC) and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE), New Ideas Track*, pages 496–499, Szeged, Hungary, September 7–9, 2011.

[Muş13] Kıvanç Muşlu. Integrating systematic exploration, analysis, and maintenance in software development. In *ICSE'13, Proceedings of the 35th International Conference on Software Engineering*, pages 1389–1392, San Francisco, CA, USA, May 22–24, 2013.

[NE02a] Jeremy W. Nimmer and Michael D. Ernst. Automatic generation of program specifications. In *ISSTA 2002, Proceedings of the 2002 International Symposium on Software Testing and Analysis*, pages 232–242, Rome, Italy, July 22–24, 2002.

[NE02b] Jeremy W. Nimmer and Michael D. Ernst. Invariant inference for static checking: An empirical evaluation. In *FSE 2002, Proceedings of the ACM SIGSOFT 10th International Symposium on the Foundations of Software Engineering*, pages 11–20, Charleston, SC, November 20–22, 2002.

[NEG+04] Toh Ne Win, Michael D. Ernst, Stephen J. Garland, Dilsun Kırlı, and Nancy Lynch. Using simulated execution in verifying distributed algorithms. *Software Tools for Technology Transfer*, 6(1):67–76, July 2004.

[NMS+08] Aleksandar Nanevski, Greg Morrisett, Avi Shinnar, Paul Govereau, and Lars Birkedal. Ynot: Dependent types for imperative programs. In *Proceedings of the 13th ACM SIGPLAN International Conference on Functional Programming (ICFP)*, Victoria, British Columbia, Canada, September 2008.

[NRZ+14] Chris Newcombe, Tim Rath, Fan Zhang, Bogdan Munteanu, Marc Brooker, and Michael Deardeuff. Use of formal methods at Amazon Web Services, September 2014. http://research.microsoft.com/en-us/um/people/lamport/tla/formal-methods-amazon.pdf.

[NYT11] NYTimes. Amazon's trouble raises cloud computing doubts, April 2011. http://www.nytimes.com/2011/04/23/technology/23cloud.html.

[OG76] Susan Owicki and David Gries. An axiomatic proof technique for parallel programs i. *Acta Inf.*, 6(4):319–340, December 1976.

[Ong14] Diego Ongaro. *Consensus: Bridging Theory and Practice*. PhD thesis, Stanford University, August 2014.

[OO14] Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm. In *Proceedings of the 2014 USENIX Annual Technical Conference*, pages 305–319, Philadelphia, PA, June 2014.

[PCG+15] Benjamin C. Pierce, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Cătălin Hriţcu, Vilhelm Sjoberg, and Brent Yorgey. *Software Foundations*. Electronic textbook, 2015.

[Pea] The peacoq development environment. http://goto.ucsd.edu/peacoq/. Accessed: 2015-09-20.

[Pet77] James L. Peterson. Petri nets. *ACM Computing Surveys*, pages 223–252, September 1977.

[PIA+07] Michael Piatek, Tomas Isdal, Thomas E. Anderson, Arvind Krishnamurthy, and Arun Venkataramani. Do incentives build robustness in BitTorrent? In Hari Balakrishnan and Peter Druschel, editors, *NSDI 2007: Proceedings of the 4th Symposium on Networked Systems Design and Implementation*, Cambridge, Massachusetts, USA, April 2007. USENIX.

[PJZ+14] Simon Peter, Umar Javed, Qiao Zhang, Doug Woos, Thomas E. Anderson, and Arvind Krishnamurthy. One tunnel is (often) enough. In Fabián E. Bustamante, Y. Charlie Hu, Arvind Krishnamurthy, and Sylvia Ratnasamy, editors, *SIGCOMM 2014: Proceedings of the ACM SIGCOMM 2014 Conference*, pages 99–110, Chicago, IL, USA, August 17-22, 2014. ACM.

[PKC+12] Lucian Popa, Gautam Kumar, Mosharaf Chowdhury, Arvind Krishnamurthy, Sylvia Ratnasamy, and Ion Stoica. FairCloud: sharing the network in cloud computing. In Lars Eggert, Jörg Ott, Venkata N. Padmanabhan, and George Varghese, editors, *SIGCOMM 2012: Proceedings of the ACM SIGCOMM 2012 Conference*, pages 187–198, Helsinki, Finland, August 13 - 17, 2012. ACM.

[PKRS11] Lucian Popa, Arvind Krishnamurthy, Sylvia Ratnasamy, and Ion Stoica. FairCloud: sharing the network in cloud computing. In Hari Balakrishnan, Dina Katabi, Aditya Akella, and Ion Stoica, editors, *HotNets 2011: Proceedings of the Tenth ACM Workshop on Hot Topics in Networks*, Cambridge, MA, USA, November 14 - 15, 2011. ACM.

[Raf] The Raft consensus algorithm. http://raft.github.io/. Accessed: 2015-09-20.

[Rah12] Vincent Rahli. Interfacing with proof assistants for domain specific programming using EventML. In *Proceedings of the 10th International Workshop On User Interfaces for Theorem Provers*, Bremen, Germany, July 2012.

[Rid09] Thomas Ridge. Verifying distributed systems: The operational approach. In *Proceedings of the 36th ACM Symposium on Principles of Programming Languages (POPL)*, pages 429–440, Savannah, GA, January 2009.

[RLH06] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271, Network Working Group, January 2006.

[RRJ+14] Daniel Ricketts, Valentin Robert, Dongseok Jang, Zachary Tatlock, and Sorin Lerner. Automating formal proofs for reactive systems. In *Proceedings of the 2014 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pages 452–462, Edinburgh, UK, June 2014.

[SE98] Chengzheng Sun and Clarence Ellis. Operational transformation in real-time group editors: Issues, algorithms, and achievements. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, CSCW '98, pages 59–68, New York, NY, USA, 1998. ACM.

[SHS+12] Justine Sherry, Shaddi Hasan, Colin Scott, Arvind Krishnamurthy, Sylvia Ratnasamy, and Vyas Sekar. Making middleboxes someone else's problem: network processing as a cloud service. In *SIGCOMM 2012: Proceedings of the ACM SIGCOMM 2012 Conference*, pages 13–24, August 2012.

[Sla10] Daniel Slane. 2010 report to Congress of the U.S.–China Economic and Security Review Commission, September 2010.

[SNB15a] Ilya Sergey, Aleksandar Nanevski, and Anindya Banerjee. Mechanized verification of fine-grained concurrent programs. In *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation, Portland, OR, USA, June 15-17, 2015*, pages 77–87, 2015.

[SNB15b] Ilya Sergey, Aleksandar Nanevski, and Anindya Banerjee. Specifying and verifying concurrent algorithms with histories and subjectivity. In *Programming Languages and Systems - 24th European Symposium on Programming, ESOP 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, pages 333–358, 2015.

[SPM+14] Adrian Sampson, Pavel Panchekha, Todd Mytkowicz, Kathryn S. McKinley, Dan Grossman, and Luis Ceze. Expressing and verifying probabilistic assertions. In *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI '14, pages 112–122, New York, NY, USA, 2014. ACM.

[SRvR+14] Nicolas Schiper, Vincent Rahli, Robbert van Renesse, Mark Bickford, and Robert L. Constable. Developing correctly replicated databases using formal tools. In *Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 395–406, Atlanta, GA, June 2014.

[SS05] Yasushi Saito and Marc Shapiro. Optimistic replication. *ACM Comput. Surv.*, 37(1):42–81, March 2005.

[SS10] Antonis Stampoulis and Zhong Shao. Veriml: Typed computation of logical terms inside a language with effects. In *Proceedings of the 15th ACM SIGPLAN International Conference on Functional Programming*, ICFP '10, pages 333–344, New York, NY, USA, 2010. ACM.

[SW01] Davide Sangiorgi and David Walker. *PI-Calculus: A Theory of Mobile Processes*. Cambridge University Press, New York, NY, USA, 2001.

[TL10] Zachary Tatlock and Sorin Lerner. Bringing extensibility to verified compilers. In *Proceedings of the 2010 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, Toronto, Canada, June 2010.

[TLSS10] Daniel Turner, Kirill Levchenko, Alex C. Snoeren, and Stefan Savage. California fault lines: Understanding the causes and impact of network failures. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, pages 315–326, New York, NY, USA, 2010. ACM.

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. http://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

[Upr] Uproxy.org. http://uproxy.org/.

[VPEJ15] Mohsen Vakilian, Amarin Phaosawasdi, Michael D. Ernst, and Ralph E. Johnson. Cascade: A universal programmer-assisted type qualifier inference tool. In *ICSE'15, Proceedings of the 37th International Conference on Software Engineering*, Florance, Italy, May 20–22, 2015.

[WLZ+14] Xi Wang, David Lazar, Nickolai Zeldovich, Adam Chlipala, and Zachary Tatlock. Jitk: A trustworthy in-kernel interpreter infrastructure. In *Proceedings of the 11th Symposium on Operating Systems Design and Implementation (OSDI)*, pages 33–47, Broomfield, CO, October 2014.

[WWP+15a] James R. Wilcox, Doug Woos, Pavel Panchekha, Zachary Tatlock, Xi Wang, Michael D. Ernst, and Thomas Anderson. Verdi: A framework for implementing and formally verifying distributed systems. In *PLDI 2015, Proceedings of the ACM SIGPLAN 2015 Conference on Programming Language Design and Implementation*, pages 357–368, Portland, OR, USA, June 15–17, 2015.

[WWP+15b] James R. Wilcox, Doug Woos, Pavel Panchekha, Zachary Tatlock, Xi Wang, Michael D. Ernst, and Thomas Anderson. Verdi: A framework for implementing and verifying distributed systems. In *PLDI 2015*, June 2015.

[YCER11] Xuejun Yang, Yang Chen, Eric Eide, and John Regehr. Finding and understanding bugs in C compilers. In *Proceedings of the 2011 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pages 283–294, San Jose, CA, June 2011.

[YCW+09] Junfeng Yang, Tisheng Chen, Ming Wu, Zhilei Xu, Xuezheng Liu, Haoxiang Lin, Mao Yang, Fan Long, Lintao Zhang, and Lidong Zhou. MoDist: Transparent model checking of unmodified distributed systems. In *Proceedings of the 6th Symposium on Networked Systems Design and Implementation (NSDI)*, pages 213–228, Boston, MA, April 2009.

[YKKK08] Maysam Yabandeh, Nikola Knežević, Dejan Kostić, and Viktor Kuncak. CrystalBall: Predicting and preventing inconsistencies in deployed distributed systems. In *Proceedings of the 5th Symposium on Networked Systems Design and Implementation (NSDI)*, pages 229–244, San Francisco, CA, April 2008.

[YV02] Haifeng Yu and Amin Vahdat. Design and evaluation of a conit-based continuous consistency model for replicated services. *ACM Trans. Comput. Syst.*, 20(3):239–282, August 2002.

[Zav12] Pamela Zave. Using lightweight modeling to understand Chord. *ACM SIGCOMM Computer Communication Review*, 42(2):49–57, April 2012.

[ZDK+13] Beta Ziliani, Derek Dreyer, Neelakantan R. Krishnaswami, Aleksandar Nanevski, and Viktor Vafeiadis. Mtac: A monad for typed tactic programming in coq. In *Proceedings of the 18th ACM SIGPLAN International Conference on Functional Programming*, ICFP '13, pages 87–100, New York, NY, USA, 2013. ACM.

[ZJW+14] Sai Zhang, Darioush Jalali, Jochen Wuttke, Kıvanç Muşlu, Wing Lam, Michael D. Ernst, and David Notkin. Empirically revisiting the test independence assumption. In *ISSTA 2014, Proceedings of the 2014 International Symposium on Software Testing and Analysis*, pages 385–396, San Jose, CA, USA, July 23–25, 2014.

[ZSS+] Irene Zhang, Naveen Kr. Sharma, Adriana Szekeres, Arvind Krishnamurthy, and Dan R. K. Ports. Building consistent transactions with inconsistent replication. SOSP '15. ACM.