http://mspiegel.github.io
[firstname].m.[lastname]@gmail.com

Michael Spiegel

1 Education

Ph.D	University of Virginia, Department of Computer Science. 2011. "Cache-Conscious Concurrent Data Structures." Committee: Andrew Grimshaw (Chair), Paul F. Reynolds (Advisor), Abhi Shelat, Steven Boker, and Doug Lea.
M.C.S.	University of Virginia, Department of Computer Science. 2006. "Model Context For Simulation Composability And Reusability." Research Advisor Paul F. Reynolds.
B.A.	Swarthmore College, Bachelor of Arts with Honors. 2003. Computer Science Major. Religion Minor. Research Advisor Tia Newhall.

2 Work Experience

Master Software Engineer

Capital One (2018-present). Tysons Corner, VA. Lead individial contributor in the automated teller machine (ATM) division. Member of the research and development team. Design and implement software from the operating system to the backend application to interact with next-generation hardware. I also rotate across several teams in ATM that support the existing software stack. My role is to solve challenging technical problems in order to reduce the technical debt of the division and to serve as a role model for junior engineers. Instituted peer review, automated build pipelines and deployments, containerized build environments, and best practices for Golang software development. Taught myself proficiency in our backend banking and credit mainframes in order to automate the management of our test environment.

Master Software Engineer

Capital One (2016-2018). Tysons Corner, VA. The Technical Fellows Program at Capital One is a team of individuals who have demonstrated ability in building massive scalable systems in domains outside of finance. They apply their collective experience to transform Capital One into a leading information-based technology company. Fellows are valued for technical knowledge and judgment; creative problem solving and innovation; technical leadership, advisory, and consulting; capability as a teacher and mentor; and technical vision. Fellows serve as role models for other individual contributors by assisting to build the deliverables. We learn by doing and by teaching others.

Software Engineer AddThis (2012-2016). Vienna, VA. Software engineer in the platform engineering team. The platform team is the expert team responsible for distributed systems within AddThis. They maintain Hydra, an open source streaming data processing system that is designed to transform records into nested data aggregates for interactive querying. It is similar in features to Apache Drill. Wrote a highly-concurrent sorted order paged object cache that spills to disk that is one of the core data structures within Hydra. Principle evangelist and author of testing and documentation to improve stability and usability. Instituted nightly automated testing. Wrote the Hydra reference documentation. Daily responsibilities include performance optimization, development of incremental feature improvements, development of long-term large feature improvements, devops application maintenance, and feedback, teaching, and guidance to other teams using our platform. Contributor to the AddThis open source projects hydra, stream-lib, metrics-reporter-config as well as several external open source projects such Google Guava. Principle author of ssync, hydra-tutor, hermes, and cronus.

Post Doc Renaissance Computing Institute (2011-2012). Chapel Hill, NC. An interdisciplinary position sponsored by the high performance computing and bioinformatics groups. The objective is the application of doctoral experience on concurrent algorithms and data structures to high performance, high throughput

	analysis of genetic information. Involved in the implementation of an efficient hierarchical OpenMP runtime, an extension of the Qthreads library (`www.cs.sandia.gov/qthreads`).
Graduate Student	Modeling and Simulation Laboratory (2006-2011). Charlottesville, VA. My doctoral research explored novel lock-free cache-conscious data structures, namely lock-free data structures that maintain the abstraction of a linearizable ordered set. I created the dense skip tree as a variation of the skip tree data structure, and proved cache-conscious properties of the dense skip tree. The lock-free dense skip tree has relaxed structural properties that allow atomic operations to modify the tree without invalidating the consistency of the data structure. Available at `www.github.com/mspiegel/lockfreeskiptree`. I show that cache-conscious, linearizable concurrent data structures have advantageous performance that can be measured across multiple architecture platforms on both synthetic and application benchmarks. Experience with reasoning about the Java virtual machine memory model and atomic operations.
General Manager	WSRN 91.5 FM (2001-2003). Swarthmore College Radio. Responsible for the continued operation of the campus 110 watt radio station. The station broadcasts daily from 9 am to 4 am, with a staff of 120 students and twelve musical and technical directors. Annual budget of approximately $8,000. Responsible for the maintenance of equipment, the training of DJs, and the compliance with FCC regulations. In my tenure I instituted regular office hours for the station staff, implemented new security policies to protect our equipment, and facilitated the rewriting of our organization's charter.

3 Technical Skills

Experienced in C (since 1995), C++ (1995), Java 1.4 - 8.0 (1999), Perl (1999), Scheme (1999), SQL (2001), Python (2006), R (2007), MPI (2011), Javascript (2014), Scala (2014), Kafka (2014), Cassandra (2014), Spark (2015), Go (2016), AWS (EC2, ECS, S3, RDS, IAM, lambda). UNIX/Linux system administration experience in server and workstation environments such as Solaris, Debian, Fedora, Centos, and Ubuntu (since 1999).

Selected Publications

Chris Bizon, Michael Spiegel, Scott A Chasse, Ian R Gizer, Yun Li, Ewa P Malc, Piotr A Mieczkowski, Josh K Sailsbery, Xiaoshu Wang, Cindy L Ehlers, et al. Variant calling in low-coverage whole genome sequencing of a native american population sample. *BMC genomics*, 15(1):85, 2014.

Stephen L. Olivier, Allan K. Porterfield, Kyle B. Wheeler, Michael Spiegel, and Jan F. Prins. OpenMP task scheduling strategies for multicore NUMA systems. *International Journal of High Performance Computing Applications*, 26(2):110–124, May 2012. URL `http://hpc.sagepub.com/content/26/2/110`. [77 citations].

Michael Spiegel. *Cache-conscious concurrent data structures*. PhD thesis, University of Virginia, 2011. URL `http://mspiegel.github.io/publications/michael-spiegel-dissertation.pdf`.

Steven Boker, Michael Neale, Hermine Maes, Michael Wilde, Michael Spiegel, Timothy Brick, Jeffrey Spies, Ryne Estabrook, Sarah Kenny, Timothy Bates, Paras Mehta, and John Fox. OpenMx: An open source extended structural equation modeling framework. *Psychometrika*, 76(2):306–317, 2011. URL `http://www.springerlink.com/content/dg37445107026711/`. [803 citations].

Michael Spiegel and Paul F. Reynolds, Jr. Lock-free multiway search trees. In *Proceedings of the 39th Annual International Conference on Parallel Processing (ICPP)*, San Diego, CA, September 13-16 2010. IEEE Computer Society. URL `http://dx.doi.org/10.1109/ICPP.2010.68`.

Tia Newhall, Sean Finney, Kuzman Ganchev, and Michael Spiegel. Nswap: A network swap module for linux clusters. In *Proceedings of the 9th European Conference on Parallel Processing (Euro-Par)*, Springer Lecture Notes in Computer Science, pages 1160–1169, 2003. [70 citations].