Joseph (Joe) Izraelevitz

joseph.izraelevitz@colorado.edu
https://www.colorado.edu/ecee/joseph-joe-izraelevitz

Research Interests

I work at the intersection of ***systems software*** and ***distributed computing theory***, and most of my work is tied together by the twin themes of ***nonvolatile memory technologies*** and ***parallel programming***. On the systems side, I have built run-time libraries for ensuring consistent state on machines with new nonvolatile memory technologies, concurrent and persistent data structures, and novel consensus protocols. On the theory side, I have proven my systems work correct and developed formal tools for reasoning about programs in nonvolatile memory. In general, I build practical systems with formal guarantees, and publish in both systems (e.g., ASPLOS, MICRO) and theory (e.g., DISC) venues.

Education

2012-2018 Ph.D. in Computer Science - University of Rochester, Rochester, NY
Dissertation title: Concurrency Implications of Nonvolatile Byte-Addressable Memory [28]
Advisor: Prof. Michael L. Scott, Dept. of Computer Science

2004-2009 B.S./M.S. in Computer Science - Washington University in St. Louis
Second major in History
Thesis title: Automated Archaeological Survey of Ancient Irrigation Canals [31]
Advisor: Prof. Robert Pless, Dept. of Computer Science

Work Experience

Aug 2019-Present University of Colorado, Boulder, CO
Assistant Professor, Electrical, Computer, and Energy Engineering

My research addresses the entire system stack in preparation for nonvolatile memory. The introduction of NVM creates both opportunities and problems. Opportunities, because the technology promises fast durable storage, low-power machine modes, and low overhead checkpointing. Problems, because traditional machine architectures, operating systems, compilers, and system libraries are not built to manage or leverage nonvolatile main memory. Serious work remains to be done in adapting the system stack to NVM if we are to realize the promise of this technology.

Aug 2018-Aug 2019 UC San Diego, San Diego, CA
Postdoctoral Scholar, Mentor: Prof. Steven Swanson

Nonvolatile Systems
The Nonvolatile Systems Lab at UCSD is a research group investigating the impact of new nonvolatile memory on the design of computer systems. Various projects exploring file systems, data structures, and transactional systems were researched, including a distributed file system accepted to FAST [4] and a novel algorithm for enforcing persistence orderings accepted to PODC [3].

Jan 2018-Aug 2018 IMDEA Software Institute, Madrid, Spain
Postdoctoral Researcher, Mentor: Dr. Alexey Gotsman

Primitives for RDMA
In contrast to event based network protocols, RDMA allows us to directly access the memory of a remote machine, bypassing the remote CPU. This project investigated how to leverage RDMA primitives in order to build extremely fast and provably correct consensus algorithms in the style of Paxos or Zookeeper's ZAB [36].

Aug 2012-Jan 2018 University of Rochester, Computer Science Department, Rochester, NY
Research Assistant, Advisor: Prof. Michael L. Scott

Infrastructure for Nonvolatile Memory
The replacement of DRAM with nonvolatile memory technologies (NVM) promises to create a new opportunity for fast, durable storage. However, at least in the near future, caches and registers are expected to remain volatile, meaning that their contents will be lost in the event of a crash. Consequently, programmers must carefully reason about the state of NVM in order to ensure that persistent data is consistent after a crash. In this line of work, I have built practical and theoretical techniques to allow application programmers to use and reason about NVM. In particular, I have proposed extremely relaxed semantic models that map easily onto all existing proposals, developed a composable correctness condition called *durable linearizability*, and proved that a simple transform exists to take any linearizable nonblocking data structure and create a durably linearizable analog that stores its state in NVM [19, 12]. On the practical side, I have built new data structures that meet the new correctness condition [9], created a method to compose durable transactions on these data structures [16], and built failure atomicity systems for crash-consistent updates [5].

Shared Memory Synchronization
In large shared-memory machines, synchronization and communication between threads is often a major obstacle to scalability. In order to improve the run-time performance of parallel programs, I investigated and designed synchronization primitives and concurrent data structures, with a particular focus on nonblocking algorithms. This work produced new nonblocking concurrent data structures [38, 7, 10] and garbage collection techniques [6], along with hardware primitives to improve the performance of locks and barriers [8].

June 2015-Aug 2016 Hewlett-Packard (HP) Labs, Palo Alto, CA
Research Associate, Systems Group, Mentor: Dr. Terence Kelly

Nonvolatile Memory and Persistent Caches
This work studied the implications that using NVM caches, in addition to NVM main memory, would have on system design (registers were assumed to remain transient). For this project, I built a high-performance failure atomicity system for high transaction volume and high concurrency [11]. This internship resulted in three patent applications related to system infrastructure for persistent caches [23, 24, 25].

June 2014-Sep 2014 Oracle Labs, Burlington, MA
Research Intern, Scalable Synchronization Group, Mentors: Dr. Yossi Lev, Dr. Virendra Marathe

Side-effects of Lock Elision
This project investigated the performance effects of lock elision using both optimistic software synchronization techniques and hardware transactional memory on real-world benchmarks [18]. The lessons learned from this project were later used to develop new hardware primitives to accelerate locks and barriers [8].

May 2009-Aug 2012 United States Army, Fort Carson, CO
Armor Officer

My three year service in the United States Army included one year-long deployment to Afghanistan as a staff officer. During my service, I filled positions in the battalion communications and planning groups, as well serving as a Tank Platoon Leader. While deployed, I served as the Battalion Afghan National Security Forces Liaison Officer, coordinating American logistical and training support for partnered security forces in Kandahar City.

Publications

Conference and Journal Publications

[1] Amirsaman Memaripour, Joseph Izraelevitz, and Steven Swanson. Pronto: Easy and fast persistence for volatile data structures. ASPLOS'20, 2020.

[2] Jian Yang, Juno Kim, Morteza Hoseinzadeh, Joseph Izraelevitz, and Steven Swanson. An empirical guide to the behavior and use of scalable persistent memory. *In: 18th USENIX Conf. on File and Storage Technologies*, FAST'20, Santa Barbara, CA, USA, 2020.

[3] Kunal Korgaonkar, Joseph Izraelevitz, Jishen Zhao, and Steven Swanson. Vorpal: Vector clock ordering for large persistent memory systems. *In: 2019 ACM Symp. on Principles of Distributed Computing*, PODC '19, Toronto ON, Canada, 2019.

[4] Jian Yang, Joseph Izraelevitz, and Steven Swanson. Orion: A distributed file system for non-volatile main memories and RDMA-capable networks. *In: 17th USENIX Conf. on File and Storage Technologies*, FAST '19, Boston, MA, 2019.

[5] Qingrui Liu, Joseph Izraelevitz, Se Kwon Lee, Michael L. Scott, Sam H. Noh, and Changhee Jung. Ido: Compiler-directed failure atomicity for nonvolatile memory. *In: 51st IEEE/ACM Intl. Symp. on Microarchitecture*, MICRO '18, Fukuoka, Japan, 2018.

[6] Haosen Wen, Joseph Izraelevitz, Wentao Cai, H. Alan Beadle, and Michael L. Scott. Interval based memory reclamation. *In: 23rd ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming*, PPoPP '18, Vienna, Austria, 2018.

[7] Joseph Izraelevitz and Michael L. Scott. Generality and speed in nonblocking dual containers. *In: ACM Trans. on Parallel Computing*, 3(4):22:1–22:37, 2017.

[8] Joseph Izraelevitz, Lingxiang Xiang, and Michael L. Scott. Performance improvement via always-abort HTM. *In: 26th Intl. Conf. on Parallel Architectures and Compilation Techniques*, PACT '17, Portland, OR, USA, 2017.

[9] Faisal Nawab, Joseph Izraelevitz, Terence Kelly, Charles B. Morrey, Dhruva Chakrabarti, and Michael L. Scott. Dalí: A periodically persistent hash map. *In: 31st Intl. Symp. on Distributed Computing*, DISC '17, Vienna, Austria, 2017.

[10] Matthew Graichen, Joseph Izraelevitz, and Michael L. Scott. An unbounded nonblocking double-ended queue. *In: 45th Intl. Conf. on Parallel Processing*, ICPP '16, Philadelphia, PA, USA, 2016.

[11] Joseph Izraelevitz, Terence Kelly, and Aasheesh Kolli. Failure-atomic persistent memory updates via JUSTDO logging. *In: 21st Intl. Conf. on Architectural Support for Programming Languages and Operating Systems*, ASPLOS'16, Atlanta, GA, USA, 2016.

[12] Joseph Izraelevitz, Hammurabi Mendes, and Michael L. Scott. Linearizability of persistent memory objects under a full-system-crash failure model. *In: 30th Intl. Conf. on Distributed Computing*, DISC '16, Paris, France, 2016.

Short Peer-reviewed Publications (Workshop papers, posters, etc.)

[13] Amirsaman Memaripour, Yi Xu, Joseph Izraelevitz, and Steven Swanson. Poster presentation: Nvhooks: Compiler support for non-volatile memory programming. NVMW '20, 2020.

[14] Jian Yang, Juno Kim, Morteza Hoseinzadeh, Joseph Izraelevitz, and Steven Swanson. An empirical guide to the behavior and use of scalable persistent memory. *In: 11th Annual Non-Volatile Memories Wkshp.* NVMW '20, San Diego, CA, USA, 2020.

[15] Faisal Nawab, Joseph Izraelevitz, Terence Kelly, Charles B. Morrey, Dhruva Chakrabarti, and Michael L. Scott. Dalí: A periodically persistent hash map. *In: 9th Annual Non-Volatile Memories Wkshp.* NVMW '18, San Diego, CA, USA, 2018.

[16] Joseph Izraelevitz, Virendra Marathe, and Michael L. Scott. Poster presentation: Composing durable data structures. *In: 8th Annual Non-Volatile Memories Wkshp.* NVMW '17, San Diego, CA, USA, 2017.

[17] Joseph Izraelevitz, Lingxiang Xiang, and Michael L. Scott. Performance improvement via always-abort HTM. *In: 12th ACM SIGPLAN Wkshp. on Transactional Computing*, TRANSACT '17, Austin, TX, USA, 2017.

[18] Joseph Izraelevitz, Alex Kogan, and Yossi Lev. Implicit acceleration of critical sections via unsuccessful speculation. *In: 11th ACM SIGPLAN Wkshp. on Transactional Computing*, TRANSACT '16, Barcelona, Spain, 2016.

[19] Joseph Izraelevitz, Hammurabi Mendes, and Michael L Scott. Brief announcement: Preserving happens-before in persistent memory. *In: 28th ACM Symp. on Parallelism in Algorithms and Architectures*, SPAA'16, Asilomar Beach, CA, USA, 2016.

[20] Joseph Izraelevitz and Michael L. Scott. Brief announcement: A generic construction for nonblocking dual containers. *In: 2014 ACM Symp. on Principles of Distributed Computing*, PODC '14, Paris, France, 2014.

[21] Joseph Izraelevitz and Michael L. Scott. Brief announcement: Fast dual ring queues. *In: 26th ACM Symp. on Parallelism in Algorithms and Architectures*, SPAA '14, Prague, Czech Republic, 2014.

Patents and Patent Applications

[22] Virendra Marathe and Joseph Izraelevitz. Systems and methods for constructing composable persistent data structures, Patent US10007581B2, US, 2018. Oracle International Corporation.

[23] Terence Kelly, Charles B. Morrey, Dhruva Chakrabarti, Aasheesh Kolli, Qiong Cai, Andrew C. Walton, and Joseph Izraelevitz. Register store, Patent application filed, US, 2016. Hewlett Packard Enterprise.

[24] Faisal Nawab, Joseph Izraelevitz, Terence Kelly, Charles B. Morrey, and Dhruva Chakrabarti. Memory system to access uncorrupted data, Patent application filed, US, 2016. Hewlett Packard Enterprise.

[25] Joseph Izraelevitz, Terence Kelly, Aasheesh Kolli, and Charles B. Morrey. Resuming execution in response to failure, Patent application filed (WO2017074451), US, 2015. Hewlett Packard Enterprise.

Unrefereed Publications (TRs, theses, etc.)

[26] Joseph Izraelevitz, Jian Yang, Lu Zhang, Amirsaman Memaripour, Yun Joon Soh, Subramanya R. Dulloor, Jishen Zhao, Juno Kim, Xiao Liu, Zixuan Wang, Yi Xu, and Steven Swanson. Basic performance measurements of the intel optane dc persistent memory module. *In: arXiv preprint arXiv:1903.05714*, 2019.

[27] Jian Yang, Juno Kim, Morteza Hoseinzadeh, Joseph Izraelevitz, and Steven Swanson. An empirical guide to the behavior and use of scalable persistent memory. *In: arXiv preprint arXiv:1908.03583*, 2019.

[28] Joseph Izraelevitz. Concurrency implications of nonvolatile byte-addressable memory, Department of Computer Science, University of Rochester, 2018. Ph.D. Thesis.

[29] Joseph Izraelevitz and Michael L. Scott. A generic construction for nonblocking dual containers. Technical report TR 992, Department of Computer Science, University of Rochester, 2014.

[30] Joseph Izraelevitz and Michael L. Scott. Fast dual ring queues. Technical report TR 990, Department of Computer Science, University of Rochester, 2014.

[31] Joseph Izraelevitz. Automated archaeological survey of ancient irrigation canals, Department of Computer Science, Washington University in St. Louis, 2009. Master's Thesis.

[32] Joseph Izraelevitz. Poster presentation: Analyzing software dependencies on supercomputers with REV. *In: Los Alamos National Laboratory Student Symp.* Los Alamos, NM, USA, 2007.

Manuscripts in Progress

[33] Juno Kim, Yun Joon Soh, Joseph Izraelevitz, Jishen Zhao, and Steven Swanson. Subzero: Zero-copy io for persistent main memory file systems, 2020. Under review at USENIX ATC.

[34] Zhang Liu, Sangtae Ha, Joseph Izraelevitz, and Dirk Grunwald. Rompaxos: Reliable ordered multicast for consensus. 2020. Planned submission to SOSP 2020.

[35] Jian Yang, Joseph Izraelevitz, and Steven Swanson. Filemr: Rethinking rdma networking for scalable persistent memory. NSDI'20, 2020. conditionally accepted.

[36] Joseph Izraelevitz, Yuri Meshman, Alexey Gotsman, and Gregory Chockler. Acuerdo: Fast atomic broadcast over RDMA, 2019. Planned submission to ASPLOS 2020.

[37] Amirsaman Memaripour, Yi Xu, Joseph Izraelevitz, and Steven Swanson. Nvhooks: Compiler support for non-volatile memory programming, 2019. In progress.

[38] Joseph Izraelevitz, Wentao Cai, Haosen Wen, Terence Kelly, Hideaki Kimura, and Michael L. Scott. Polytree: A synchronization framework for building ordered maps, 2018. In progress.

Teaching Experience

Mentored Students

Undergraduate and junior graduate students at the Univ. of Rochester have the opportunity, as part of some seminar courses, to work on independent research projects with senior graduate students. Often, these projects turn into full-fledged research collaborations that continue after the end of the class, generally with minimal faculty supervision. Through this arrangement, I have been the primary research mentor for three students.

- *Wentao Cai*, Ph.D. at Univ. of Rochester (2016-2017)
  Project: A framework for building nonblocking trees [38] (in progress).
- *Haosen Wen*, Ph.D. at Univ. of Rochester (2016-2017)
  Project: Interval Based Memory Reclamation
  This project resulted in a full paper at the Symp. on Principles and Practice of Parallel Programming (PPoPP) [6].
- *Matthew Graichen*, B.S. at Univ. of Rochester (2014-2016)
  Project: Unbounded Obstruction-Free Deques
  This year-long project resulted in a full paper at the Intl. Conf. on Parallel Programming (ICPP) [10].

Teaching

- *Instructor, C Programming. (SP2020)* This introductory programming class covers the essentials of the C and C++ languages for engineering majors.
- *Instructor, Concurrent Programming. (FL2019)* This class covers the essentials of concurrent programming, covering both shared memory synchronizationand distributed systems.
- *Teaching Assistant for two semesters of CS200 (Undergraduate Research Seminar).* This class exposes undergraduates to the process of doing research. During the class, the students, in groups, complete a couple of research projects on open problems in computer science. As a TA, I was responsible for guiding these groups as they chose their research problems and for mentoring them as they worked their way towards solutions. I also gave two lectures on my own research, and was responsible for the normal TA duties (grading, attendance, etc.).
- *Teaching Assistant for CS486 (Computational Complexity).* This class is an in-depth exploration of complexity theory at the graduate level and is a required course for Ph.D. students, who normally take it in their first year. As this class is often students' first exposure to difficult ideas in theoretical computer science, the TA office hours were well attended. While the primary instructor was out, I gave a series of three full lectures on class material, covering the Chomsky Hierarchy, with its associated grammars and machines, and Rice's Theorem. I was also responsible for the normal TA duties (grading, office hours, etc.).

Honors and awards

- Commendation (Runner-up) in Outstanding Dissertation Award Competition for Engineering, Univ. of Rochester, 2018.
- Hopeman Fellowship (School of Engineering scholarship), 2013-2014.
- Sproull Fellow (full-tuition award to Univ. of Rochester), 2012-2013.
- Induction into Tau Beta Pi – Engineering Honor Society, 2009.
- Induction into Phi Alpha Theta – History Honor Society, 2009.
- George C. Marshall Cadet Award (top military graduate at Washington Univ. in St. Louis), 2009.
- Alexander S. Langsdorf Fellow (full-tuition merit-based award to Washington Univ. in St. Louis), 2004-2009.
- National Merit Finalist, 2003.

Grants

- PI for *Tools for Programmability and Performance on NVDIMMs* for the NSF Computer and Network Systems Core Program (under review).
- PI for *Persistent Atomics* for internal Intel Grant Program (under review).
- Co-PI for *Enabling High IPC in Future Multi-NUMA Systems* for the NSF/Intel Partnership on Foundational Microarchitecture Research (FoMR) Program, based on my work done in [8],[17] (under review).
- Authored equipment grant to Mellanox Technologies for joint project with IMDEA Software and Tel-Aviv University, conditionally approved for funding of approximately $30,000
- Assisted in writing CCF-1717712 ($449,937): *Data Structures and Transactions for Emerging Nonvolatile Memory* awarded an NSF Small Proposal Grant in 2017; based on my work done in [11],[12],[19],[10],[7].
- Awarded travel grants to PACT'17, SPAA'16.

Service

- *Reviewer.* Very Large Data Base (VLDB) Journal, 2020.
- *PC Member.* ACM Symp. on Parallelism in Algorithms and Architectures (SPAA), 2020.
- *PC member.* Nonvolatile Memories Workshop (NVMW), 2020.

- *PC member*. IEEE/ACM Intl. Symp. on Cluster, Cloud and Grid Computing (CCGrid), 2020.
- *Reviewer*. IEEE Computer Architecture Letters (CAL), 2019.
- *Organizing Committee*. Persistence in Real Life (PiRL), 2019.
- *External PC*. ACM Symp. on Principles of Distributed Computing (PODC), 2019.
- *Reviewer*. IEEE Trans. on Parallel and Distributed Systems (TPDS), 2019.
- *Reviewer*. ACM Trans. on Architecture and Code Optimization (TACO), 2019.
- *PC member*. Nonvolatile Memories Workshop (NVMW), 2019.
- *Publicity Chair*. Nonvolatile Memories Workshop (NVMW), 2019.
- *PC member*. IEEE/ACM Intl. Symp. on Cluster, Cloud and Grid Computing (CCGrid), 2019.
- *Co-reviewer*. IEEE/IFIP Intl. Conf. on Dependable Systems and Networks (DSN), 2019.
- *Co-reviewer*. ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming (PPoPP), 2018.
- *External PC*. Intl. Symp. on Memory Management (ISMM), 2017.
- *Artifact Evaluation PC*. ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming (PPoPP),2017.
- *External PC*. Intl. Conf. on Parallel Architectures and Compilation Techniques (PACT), 2016.
- *Co-reviewer*. IEEE Trans. on Parallel and Distributed Systems (TPDS), 2016.
- *Co-reviewer*. ACM Symp. on Parallelism in Algorithms and Architectures (SPAA), 2016.
- *Co-reviewer*. Intl. Symp. on Computer Architecture (ISCA), 2016.
- *Co-reviewer*. ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming (PPoPP), 2016.
- *Co-reviewer*. Intl. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2016.
- *External PC*. Intl. Conf. on Distributed Computing (DISC), 2014.

Talks

- Invited talk at Intl. Wkshp. on High Performance Transaction Systems (HPTS): An Empirical Guide to the Behavior of Scalable Persistent Memory, 2019.
- Invited talk at Microsoft Research: Distributed Algorithms over RDMA. Cambridge, UK. July, 2018
- Invited talk at VMWare Research: Linearizability of Persistent Memory Objects Under a Full-System-Crash Failure Model. Palo Alto, CA, August 2016.
- Presented papers at ASPLOS [11], DISC×2 [12],[9], PACT [8], SPAA×2 [19],[21], TRANSACT×2 [17],[18].