Vaibhav Gogte

vgogte@umich.edu | +1 (352) 870-7970 | https://vgogte.github.io

Research Interests

My current research focuses on designing runtime systems and hardware architecture for emerging byte-addressable persistent memories. Specifically, I have designed memory models in high-level languages and processor architectures that allow easier persistent memory programming, and built software systems that use persistent memories as faster storage.

Education

- **Ph.D. Candidate, Computer Science and Engineering** Sept 2014 – Dec 2019
  M.S. Computer Science and Engineering Sept 2014 – Apr 2016
  University of Michigan - Ann Arbor (GPA: 4.0/4.0)
  Thesis: Runtime Systems for Persistent Memories
  Advisor: Prof. Thomas F. Wenisch
- **B.E. (Hons.), Electrical & Electronics Engineering** Aug 2007 – May 2011
  Birla Institute of Technology & Science - Pilani, India (GPA: 8.92/10.0)

Experience

Software Engineer, *Google LLC* Mar 2020 – Present
Research Fellow, *University of Michigan, Ann Arbor* Jan 2020 – Mar 2020

- **Primer on memory persistency**: I worked on a synthesis lecture for memory persistency, that covers the state-of-the-art research on architectural and software support for persistent memories.

Graduate Student Research Assistant, *University of Michigan, Ann Arbor* Sept 2014 – Dec 2019

- **Strand Persistency**: In this work, I proposed hardware primitives to minimally constrain ordering on persistent memory operations. I designed compiler mechanisms that map persistency semantics in high-level languages to the hardware primitives. This work achieves up to 1.97× speedup over existing hardware ordering mechanisms.
- **Persistency for Synchronization-Free Regions**: In this work, I proposed persistency semantics that guarantee failure atomicity of synchronization-free regions, program regions delimited by synchronization operations. Our approach provides clear semantics for post-failure state by extending sequentially consistent guarantees to the recovery code.
- **Language-Level Persistency**: This project defines an acquire-release persistency model as an extension to the C++11 memory model. The persistency model relaxes the ordering constraints while accessing recoverable data structures in persistent memory-enabled systems.
- **OS-based Wear Management for Persistent Memories**: Emerging persistent memories suffer from a limited write endurance. I designed an application-transparent wear-management technique in Linux kernel that detects the disparity in the write accesses and performs targeted page migrations to manage wear of the persistent memory.
- **Hardware Accelerator for Regular Expressions**: In this work, I designed a stall-free hardware accelerator for matching regular expressions at a scan rate offered by modern DDRs. I built a compiler toolchain in C++ that transforms the regular expressions to the binary used by our hardware design.
- **Programmable and Energy-Efficient 3D Convolution Engine**: I developed a hardware accelerator that improves data locality when fetching 3D images from the main memory. The convolution engine overlaps the processing of locally cached image pixels with the fetching of a new set of pixel data from the main memory.

Performance and Capacity Research Intern, *Facebook, Menlo Park* May 2019 – Aug 2019

- **Persistent Memory Usecases for Facebook Applications**: In this work, I designed persistent memory-aware block cache for RocksDB, a key-value store application. I evaluated several RocksDB usecases in Facebook applications to show throughput and latency improvement due to the persistent memory-aware cache.

Research Intern, *Microsoft Research, Redmond* May 2018 – Aug 2018, May 2017 – Aug 2017

- **Low-cost and Predictable Storage Management for Persistent Memories**: The access latency of persistent memories, storage features and modern networks are a few microseconds. In this work, I proposed low-cost and predictable storage by offloading their management to the ARM-based commodity SoC device attached to the host over PCIe.

- **Load-balancing in Catapult architecture**: I extended the routing and transport layer in the Catapult architecture to dynamically balance query load in the FPGA-enabled servers. The proposed mechanism monitors the response latency and manages the load offered by the FPGAs.

Senior Design Engineer, *Texas Instruments, Bangalore* Jul 2011 – Jul 2014

- **Design of microprocessor subsystems based on ARM processors**: I designed power management, protocol converters, and interrupt controllers for dual-core microprocessor subsystems based on ARM Cortex A9, A7, and M4 processors.

Teaching Experience

Graduate Student Instructor, *University of Michigan, Ann Arbor*

- Parallel Computer Architecture (EECS 570) (Received Honourable Mention) Jan 2017 – May 2017
- Introduction to Logic Design (EECS 270) Jan 2015 – May 2015

Peer-Reviewed Conference Publications

- **Improving Performance of Flash Based Key-Value Stores Using Storage Class Memory as a Volatile Memory Extension**
  H. Kassa, J. Akers, M. Ghosh, Z. Cao, **V. Gogte**, R. Dreslinski.
  2021 USENIX Annual Technical Conference (**ATC**). July 2021.
- **Relaxed Persist Ordering Using Strand Persistency**
  V. Gogte, W. Wang, S. Diestelhorst, P. M. Chen, S. Narayanasamy, T. F. Wenisch.
  47th Annual International Symposium on Computer Architecture (**ISCA**). Jun 2020.
- **LeapIO: Efficient and Portable Virtual NVMe Storage on ARM SoCs**
  H. Li, M. Hao, S. Novakovic, **V. Gogte**, S. Govindan, D. R. Ports, I. Zhang, R. Bianchini, H. S. Gunawi, A. Badam.
  24th International Conf. on Arch. Support for Prog. Languages and Operating Systems (**ASPLOS**). Mar 2020.
- **Language Support for Memory Persistency**
  A. Kolli, **V. Gogte**, A. Saidi, S. Diestelhorst, W. Wang, P. M. Chen, S. Narayanasamy, T. F. Wenisch.
  Top Picks of the 2019 Computer Architecture Conferences (**Top Picks**). Jun 2019.
- **Software Wear Management for Persistent Memories**
  V. Gogte, W. Wang, S. Diestelhorst, A. Kolli, P. M. Chen, S. Narayanasamy, T. F. Wenisch.
  17th USENIX Conference on File and Storage Technologies (**FAST**). Feb 2019.
- **Persistency for Synchronization-Free Regions**
  V. Gogte, W. Wang, S. Diestelhorst, S. Narayanasamy, P. M. Chen, T. F. Wenisch.
  39th ACM SIGPLAN Conference on Programming Language Design and Implementation (**PLDI**). Apr 2018.
- **Language-Level Persistency**
  A. Kolli, **V. Gogte**, A. Saidi, S. Diestelhorst, P. M. Chen, S. Narayanasamy, T. F. Wenisch.
  44th Annual International Symposium on Computer Architecture (**ISCA**). Jun 2017.
- **HARE: Hardware Accelerator for Regular Expressions**
  V. Gogte, A. Kolli, M. J. Cafarella, L. D'Antoni, T. F. Wenisch.
  49th International Symposium on Microarchitecture (**MICRO**). Oct 2016.
- **NoCVision: A Network-on-Chip Dynamic Visualization Solution**
  V. Gogte, D. Lee, R. Parikh, V. Bertacco.
  8th International Workshop on Network on Chip Architectures (**NoCArc**). Dec 2015.

Peer-Reviewed Workshop Publications

- **Hardware Implementation of Strand Persistency**
  V. Gogte, W. Wang, S. Diestelhorst, P. M. Chen, S. Narayanasamy, and T. F. Wenisch.
  11th Annual Non-Volatile Memories Workshop (**NVMW**). Mar 2020.
- **Strand Persistency**
  V. Gogte, W. Wang, S. Diestelhorst, P. M. Chen, S. Narayanasamy, and T. F. Wenisch.
  10th Annual Non-Volatile Memories Workshop (**NVMW**). Mar 2019.
- **Failure-Atomic Synchronization-Free Regions**
  V. Gogte, W. Wang, S. Diestelhorst, S. Narayanasamy, P. M. Chen, and T. F. Wenisch.
  9th Annual Non-Volatile Memories Workshop (**NVMW**). Mar 2018.

- **TARP: Translating Acquire-Release Persistency**
  A. Kolli, **V. Gogte**, A. Saidi, S. Diestelhorst, P. M. Chen, S. Narayanasamy, and T. F. Wenisch.
  8th Annual Non-Volatile Memories Workshop (**NVMW**). Mar 2017.

Books

- **A Primer on Memory Persistency** (Under preparation)
  V. Gogte, A. Kolli and T. F. Wenisch.
  Morgan and Claypool Publishers

Press

- Three CSE papers chosen as IEEE Micro Top Picks [Link], *The Michigan Engineer*, 2019
- Persistency for Synchronization-Free Regions [Link], *Arm Research*, 2018
- Baking Specialization into Hardware Cools CPU Concerns [Link], *Next Platform*, 2016

Patents

- **Instruction ordering**
  V. Gogte, W. Wang, S. Diestelhorst, P. M. Chen, S. Narayanasamy, T. F. Wenisch. (*US Patent 10,956,166*)
- **Detecting at least one predetermined pattern in stream of symbols**
  M. J. Cafarella, **V. Gogte**, T. F. Wenisch. (*US Patent 10,339,141*)

Selected Talks and Presentations

- **Relaxed Persist Ordering Using Strand Persistency**
  47th Annual International Symposium on Computer Architecture (**ISCA**). Jun 2020.
- **Hardware Implementation of Strand Persistency**
  11th Annual Non-Volatile Memories Workshop (**NVMW**). Mar 2020.
- **Runtime Systems for Persistent Memories**
  AMD Research. Oct 2019.
  ARM Review. Jan 2018.
- **Persistent Memory Usecases for Facebook Applications**
  Facebook HQ. Aug 2019.
- **Strand Persistency**
  10th Annual Non-Volatile Memories Workshop (**NVMW**). Mar 2019.
- **Software Wear Management for Persistent Memories**
  17th USENIX Conference on File and Storage Technologies (**FAST**). Feb 2019.
- **Low-cost and Predictable Storage Management for Persistent Memories**
  Microsoft Research. Aug 2018.
- **Persistency for Synchronization-Free Regions**
  39th ACM SIGPLAN Conference on Programming Language Design and Implementation (**PLDI**). Apr 2018.
- **Failure-Atomic Synchronization-Free Regions**
  9th Annual Non-Volatile Memories Workshop (**NVMW**). Mar 2018.
- **Load-Balancing in Catapult Architecture**
  Microsoft Research. Aug 2017.
- **TARP: Translating Acquire-Release Persistency**
  8th Annual Non-Volatile Memories Workshop (**NVMW**). Mar 2017.
- **HARE: Hardware Accelerator for Regular Expressions**
  49th International Symposium on Microarchitecture (**MICRO**). Oct 2016.
- **NoCVision: A Network-on-Chip Dynamic Visualization Solution**
  8th International Workshop on Network on Chip Architectures (**NoCArc**). Dec 2015.

Service

- **Reviewer** for ISCA'20, NVMW'20, IEEE CAL'19, TOMPECS'19, TACO'18 and TODAES'16
- **Shadow Program Committee Member** for EuroSys'19

Technical Skills

- **Languages**: C, C++, Verilog HDL
- **Software and Design Tools**: LLVM, Murphi, DynamoRio, Git, Latex
- **Architectural simulators**: Gem5, Booksim