Vijayanand (Vijay) Nagarajan

University of Edinburgh
School of Informatics
10 Crichton Street
Edinburgh, EH8 9AB

Email: vijay.nagarajan@ed.ac.uk
URL: http://homepages.inf.ed.ac.uk/vnagaraj/
Telephone: (+44) 131-651-3440
Mobile: 07879529538

CURRENT APPOINTMENT

Reader (Associate Professor), School of Informatics, University of Edinburgh, August 2015 - present.

Honorary Visiting Faculty, Department of Computer Science and Engineering, IIT Madras, November 2018 - March 2019

EDUCATION

Ph.D. in Computer Science, University of California Riverside, September 2009.

M.S. in Computer Science, The University of Arizona, May 2005.

B.E (Bachelor of Engineering) in Computer Science, University of Madras, March 2003.

PREVIOUS APPOINTMENTS

Lecturer (Assistant Professor), School of Informatics, University of Edinburgh, October 2009 - July 2015.

Research Assistant, Dept. of Computer Science and Eng., UC Riverside, August 2007 - September 2009.

Research Intern, Programming Systems Lab, Intel Corporation, May 2006 - August 2006.

Research Assistant, Dept. of Computer Science., Univ of Arizona, May 2004 - July 2007.

AWARDS AND HONOURS

Intel 2013 Early Career Faculty Honour Award

PACT Best Paper Award: "Efficient Sequential Consistency Using Conditional Fences", PACT 2010.

IEEE MICRO Top Picks Honorable Mention: "ProtoGen: Automatically Generating Directory Cache Coherence Protocols from Atomic Specifications," ISCA 2018.

HiPEAC Paper Awards: 2018, 2017, 2016, 2015

PhD Supervisor of Bharghava Rajaram, Intel 2012 Doctoral Student Honour Award.

RESEARCH INTERESTS

My research interests span: Computer Architecture, Compiler Optimizations and Computer Systems., with a focus on: Memory Consistency Models and Cache Coherence Protocols.

RESEARCH CONTRIBUTIONS AND IMPACT

I have published 10 journal papers, 36 conference papers[1] and 8 workshop papers in top-tier venues including at Computer Architecture (ISCA, HPCA, MICRO), Programming Languages and Compilers (PLDI, PACT), High-performance computing (SC, ICS), Computer Systems (EuroSys, ASPLOS), Robotics (RSS,

[1] In my area, the top-tier conference papers are rigorously reviewed and are more prestigious than journals, https://goo.gl/4v6PJm

IROS), Verification (FMCAD), and Software Engineering (ISSTA, ICSM). I have been invited to co-write the 2nd edition of the *Primer on Memory Consistency and Cache Coherence* (Morgan & Claypool Publishers). I've been invited to the RISC-V memory model task group (RISC-V is the high-profile open source processor initiative) tasked with designing its memory consistency specification. The specification we designed has been ratified by the community and by the board of directors of the RISC-V Foundation. My research has had industrial impact: we collaborated with Intel to help shape their persistent memory primitives; we caught and fixed coherence protocols bugs in the most widely used simulator (including by ARM and AMD); our work on instruction prefetching has resulted in a patent application (two major companies are in talks with us for potential licensing). My contributions in the past decade can be summarised as follows.

(i) Shared Memory Consistency and Cache Coherence

Shared memory is an attractive parallel programming model but the illusion of shared memory is something that has to be *enforced* and comprises three components. First, the *memory consistency model* that specifies what value a read can return. Second, the *cache coherence protocol* which is a subsystem that helps enforce the memory consistency model by making caches transparent to the programmer. Third, primitive synchronisation operations such as memory fences and read-modify-writes (RMWs), using which higher-level synchronisation constructs are implemented. Unfortunately, there have been two long-standing problems associated with shared memory.

Tension between efficiency and programmability. The most intuitive memory consistency model, sequential consistency (SC), is presumed to be too expensive to support; likewise primitive synchronisation instructions like memory fences and atomic read-modify-writes (RMWs) are costly in current processors; finally, there are question marks about whether cache coherence can be made to scale with increasing number of cores.

Bugs in implementation. For instance, coherence protocol bugs in real processors (such as the one used by Samsung in its Galaxy range of mobile phones) are not uncommon. In the past decade, my research has made inroads on these long-standing problems and in the process, has overturned conventional wisdom on a number of significant ways.

Efficient SC. We were the first to show that strong consistency models such as SC can be enforced at high performance without requiring power-hungry speculation [PACT'10, ASPLOS'12]. Our key innovation is a novel way to achieve memory ordering called *conflict ordering*: instead of conservatively ordering memory operations from the same processor, we only order conflicting memory operations across different processors. Our work has lead to significant follow-up work on non-speculative techniques for achieving memory ordering (including by research groups at Illinois, Michigan and Wisconsin)

Efficient RMWs RMWs are expensive, especially on x86 architectures where they are ordered like a memory barrier. We show that this is because of the strict atomicity semantics of the RMW [PLDI'13]. We argue for x86 RMWs to use weaker atomicity definitions for significantly more efficient implementations, while showing that the relaxed RMWs can be used in synchronisation situations like before. This paper was discussed in the RISC-V memory model task group and provided clarity on RMW semantics.

Scalable coherence for TSO. We show how cache coherence can be made to scale on x86 processors with a sharing-vector-free approach to coherence [HPCA'14]. We observe that using protocols that eagerly propagate writes on systems that support memory models weaker than SC is overkill. For the first time, we tailor the (lazy) coherence protocol for TSO supported by x86 processors. We released the complete specification of the protocol (dubbed TSO-CC) to the community (https://goo.gl/2fPVg7), allowing for other research groups to implement or verify our protocol.

Coherence protocol verification. Because TSO-CC breaks the traditional coherence-consistency separation, the coherence protocol now has to be verified against the consistency model, a long-standing open problem. We first explore a testing based approach. We discovered that traditional testing (random testing and litmus testing) is too slow in a simulator environment. We propose an automatic genetic-programming based approach for better tests [HPCA'16]. In that process we discovered two bugs in a widely-used simulator (Gem5) used extensively by the academia and industry (ARM and AMD); we alerted the developers and sent patches to fix the bugs. We have also open sourced our simulator-independent library (https://goo.gl/iQ78rh)

A testing based approach, while useful, is not a proof. We propose the first formal and complete technique to verify a coherence protocol against the consistency model [FMCAD'17]. Our key innovation is identifying a novel abstraction for TSO called TSO-LB, which also captures the essential idea behind our protocol TSO-CC. After manually proving that TSO-LB satisfies TSO, we then showed that our protocol (TSO-CC) satisfies TSO automatically within a model checker. We have open-sourced our verification results (https://goo.gl/iPuY7s).

Protocol Synthesis. Identifying TSO-LB (during verification) made us introspect back to how we arrived at TSO-CC (during the research process); we realised that we had (manually) come up with some thing like TSO-LB, which we had (manually) concretised into TSO-CC. Thinking back to some of our earlier contributions—for instance efficient SC—we realised that we had employed the same *modus operandi*: (1) Transform abstract operational model into one that is fit for the purpose (for efficient SC, transforming from *program ordering* to *conflict ordering*). (2) Generate concrete implementation from the abstract operational model.

We asked ourselves whether it would be possible to automate this overarching strategy. Specifically, we asked whether it would be possible to automatically synthesise a complete coherence protocol given only a stable state protocol (SSP); SSPs are the familiar textbook protocols containing only the stable states and transitions, whereas the complete protocol includes transient states and actions which are notoriously complex. Out tool, ProtoGen [ISCA'18], takes as its input the SSP expressed in our bespoke DSL and generates a high-performance complete protocol with transient states. Our tool has been open sourced (https://goo.gl/2W8cyh)—efforts are underway to integrate it into a popular simulator.

Cache-coherent Key-value stores. Today's cloud based online services are underpinned by distributed key-value stores (KVS). One important performance bottleneck is the load imbalance caused by skewed popularity distributions. We make the key observation that in light of modern RDMA based networking, a KVS is not all that different from shared memory. We advocate for embracing popularity skew as a performance opportunity and propose aggressively caching popular items at all nodes of the KVS. To main consistency across the caches, we use fully-distributed consistency protocols, taking inspiration from shared memory coherence protocols. A key result of this work is that very strong consistency guarantees (per-key linearizability) need not compromise on performance [EuroSys'18]. We have open sourced our cache coherent key-value store (https://goo.gl/p3QVJU) as well as our verfied protocol specification (https://goo.gl/zStnPD).

(ii) Other Contributions

Architectural support for Persistent Memory. Emerging persistent memory technologies enable fast, fine-grained durability compared to slow block-based devices. Programming with persistent memory, however, requires primitives that provide guarantees about what has been made durable. Two primitives that programmers understand well are: ordering (persist barrier) and atomicity (atomic durable transactions and ACID transactions). We show for the the first time how these primitives can be realised efficiently in shared memory multiprocessors [MICRO'15, HPCA'17, ISCA'18]

Architectural support for DRAM caches. 3-D stacking technology has enabled the option of embedding a large DRAM cache onto the processor. This, however, poses a number of challenges: How to manage the tags? How to schedule tag and data accesses? How to keep large DRAM caches coherent? We address these challenges [PACT'14, SC'16, MICRO'16] paving the way for adoption of DRAM caches.

Addressing the Front-end Bottleneck. In collaboration with Boris Grot, we develop two techniques to address the well known front-end bottleneck for server programs. We propose Boomerang [HPCA'17], a metadata-free architecture that leverages a branch-predictor-directed prefetcher to discover and prefill instruction cache blocks. Unfortunately, Boomerang has limited effectiveness on workloads with frequent BTB misses. In Shotgun [ASPLOS'18], we propose a new BTB organization that enables high-efficacy prefetching at low storage cost.

CONTRIBUTIONS TO KNOWLEDGE EXCHANGE AND IMPACT

I have had significant collaboration with industry including with Intel and ARM. I have had one patent with Intel and one another that has been submitted by the Edinburgh commercialisation team (in talks with two major companies who have evinced an interest).

I collaborated with Intel on a 3-year URO funded project on persistent memory. Persistent memory is the new disruptive technology that has the potential to unify memory and storage. In collaboration with Intel, we proposed microarchitectural implementations of three such primitives: an ordering primitive (MICRO'15), atomic durability primitive (HPCA'17) and ACID transaction primitive (ISCA'18). We disclosed our findings to Intel in April 2016 suggesting an implementation in which the memory controller is logically persistent. Intel released an open disclosure consistent with our findings in Sep 2016 (deprecation of pcommit instruction).

I was invited into the the RISC-V memory model working group, tasked with designing the memory consistency model. RISC-V is a community effort for an open ISA and its associated ecosystem. This has generated huge interest both in industry as well as academia as a potential game-changer that can lead to computer architecture start-ups. Unfortunately, the original RISC-V memory model specification had bugs that rendered it inadequate to generate high-level language mappings. Consequently, a memory model task group was formed consisting of semantics, computer architecture and verification experts on memory models from the academia and industry. My research on consistency and cache coherence was discussed during the task group meetings. In particular, my work on atomicity semantics of a read-modify-writes [PLDI'13] helped the group arrive at a suitable atomicity semantics for RISC-V. Secondly, there was a technical debate on whether the RISC-V memory model should be TSO or any weaker. As part of the debate, my work and its follow-on works on non-speculative consistency enforcement were discussed. The memory model we co-designed has been recently ratified by the community and by the board of directors of the RISC-V foundation. It is worth noting that the utility of open and formal specification of memory model in this fashion goes beyond RISC-V as it serves as an exemplar for other companies to follow.

US Patent 8,321,840: "Software Flow Tracking using Multiple Threads", Inventors: V. Nagarajan, H-S. Kim (Intel), Y. Wu (Intel), and R. Gupta.

International Patent Application No PCT/GB2018/050294: "Boomerang: a Metadata-Free Architecture for Control Flow Delivery", Inventors: Rakesh Kumar, Boris Grot and Vijay Nagarajan.

Co-author of RISC-V ISA specification. https://github.com/riscv/riscv-isa-manual/releases/download/draft-20180918-3318c9e/riscv-spec.pdf

GRANTS

Leverhulme (PI) *Host of Visiting Professor: Prof. Daniel Sorin*, £24,395 , 2017.

SICSA (PI) *Host of Distinguished Visiting Fellow: Prof. Daniel Sorin*, £6080 , 2017.

EPSRC (Edinburgh PI) *C3: Scalable and verified shared memory via Consistency-directed Cache Coherence*, £668,897 , 6/2015-6/2018.

Amazon AWS Research Education Grant (PI), *Towards Efficient and Scalable Causal Consistency* $5250, 6/2015-6/2016

EPSRC (Edinburgh PI) *ESP-SD: Error tolerant Stream Processing System Design*, £441,324 , 10/2014-9/2017.

EPSRC (Edinburgh PI) *Anyscale Applications*, £ 482,610 , 9/2013-8/2017.

Intel (co-PI) *Dapper: Database inspired persistent Memory*, £174,762, 9/2013 - 8/2016 (PI: Stratis Viglas)

Intel (PI) *Early Career Faculty Honour*, £21,381, 10/2013 - 9/2014.

UKIERI (co-PI) *Power efficient and high performance data prefetching techniques for multi core processors*, £8,873, 1/2012 - 6/2014 (PI: Murray Cole)

PhD Students (graduated)

Bharghava Rajaram, 2014 (*Assistant Professor, Mahindra Ecole Centrale, India*).

Andrew McPherson (with M. Cintra), 2015 (*First: IBM Research; Current: Senior Data Scientist, Credit Suisse*).

Cheng-Chieh Huang, 2015 (*Software Engineer, Google*).

Marco Elver, 2016 (*Software Engineer, Google, Mountain View, CA*).

Saumay Dublish (with N.P. Topham), 2018 (*Synopsis, starting Sep 2018*).

Arpit Joshi, 2018 (*Researcher, Intel, Portland, OR*)

Post-docs (graduated)

Rakesh Kumar, 2017 (*Associate Professor, Norwegian University of Science and Technology)*

Jose Cano Reyes, 2018 (*Assistant Professor, Glasgow University)*

Chris Banks, 2018 (*Research Scientist, Roslin Institute and School of Veterinary studies, Edinburgh)*

PhD Students (in progress)

Vasillis Gavrielatos, expected 2020

Nicolai Oswald, expected 2021

Mahesh Dananjaya, expected 2022

INVITED PRESENTATIONS

ARM Research Summit, "Cache Coherence Protocols are Notoriously ~~Hard~~ Easy", Cambridge, UK, September 2018.

IIT Madras, "Scaling: Up and Out", Chennai, India, December 2017.

Intel Research, "Semantics-directed Hardware Design for Shared Memory", Intel Research, Bangalore, December 2017.

Dagstuhl Seminar, "Architectural Support for Persistent Memory", Dagstuhl, November 2017.

ARM Research Summit, "Scaling: Up and Out", Cambridge, UK, September 2017.

University of Pennsylvania, "Semantics-directed Hardware Design for Shared Memory", Philadelphia , USA, December 2016.

HP Labs, "Efficient Architecture Support for Persistent Memory", Palo Alto , USA, December 2016.

Princeton University, "Semantics-directed Hardware Design for Shared Memory", Princeton , USA, December 2016.

ARM Research Summit, "Semantics-directed Hardware Design for Shared Memory", Cambridge, UK, September 2016.

Newcastle University, "Hardware Support for Shared-memory Concurrency: Reconciling Programmability with Performance", Newcastle, UK, August 2016.

University of Michigan, Computer Architecture Reading Group, "Hardware Support for Shared-memory Concurrency: Reconciling Programmability with Performance", Ann Arbor, USA, June 2016.

Rutgers University, "Hardware Support for Shared-memory Concurrency: Reconciling Programmability with Performance", New Brunswick, USA, June 2016.

Queen's University Belfast, "Hardware Support for Shared-memory Concurrency: Reconciling Programmability with Performance", Belfast, UK, April 2016.

Chennai Mathematical Institute, "Hardware Support for Shared-memory Concurrency: Reconciling Programmability with Performance", Chennai, India, December 2015.

IIT Madras, Dept. of CSE, "Efficient Persist Barriers for Multicores", Madras, India, December 2015.

Huawei Research, Santa Clara, "Support for Shared-memory Concurrency: Reconciling Programmability with Performance", Santa Clara, USA, October 2015.

UPMARC Workshop of Memory Models, Uppsala, "TSO-CC: Consistency-directed Cache Coherence for TSO", Uppsala, Sweden, February 2015.

University of Glasgow, "Hardware Support for Shared-memory Concurrency: Reconciling Programmability with Performance", Glasgow, UK, October 2014.

Imperial College, London, "Hardware Support for Shared-memory Concurrency: Reconciling Programmability with Performance", London, UK, October 2014.

IISc Bangalore, "Hardware Support for Shared-memory Concurrency: Reconciling Programmability with Performance", Bangalore, India, December 2013.

Beihang University, "Hardware Support for Shared-memory Concurrency: Reconciling Programmability with Performance", Beijing, China, November 2013.

Intel Research, Santa Clara, "Hardware Support for Shared-memory Concurrency: Reconciling Programmability with Performance", Santa Clara, USA, June 2013.

Intel Research, Braunschweig, "Efficient Sequential Consistency without Aggressive Speculation", Braunschweig, Germany, May 2012.

University of Manchester, APT Talk, "Monitoring Parallel Programs for Performance and Reliability", Manchester, UK, March 2011.

IIT Madras, Dept. of CSE, "Monitoring Parallel Programs for Performance and Reliability", Madras, India, July 2010.

University of Edinburgh, NAIS Annual Meeting, "Monitoring Parallel Programs for Performance and Reliability", Edinburgh, UK, June 2010.

University of Cambridge, Hardware Discussion Series, "Enabling Runtime Monitoring on Multicores", Cambridge, UK, June 2010.

INVITED SHORT COURSE

UPMARC Summer School, "Hardware Support for Shared-memory Concurrency: Reconciling Programmability with Performance", Uppsala, Sweden, June 2016.

Beihang University, Invited Short course on "Hardware support for Shared Memory", Beijing, China, November 2013.

IIT Madras, Dept. of CSE, Invited Distinguished Short course on "Hardware Support for Shared Memory", Madras, India, December 2012.

PhD Committee: External

Albert Esteve Garcia (Polytechnic University of Valencia), 2017

Raghavendra K (IIT Madras), 2017

Robert Norton (University of Cambridge), 2015

Mohammad Ashraful Anam (University College London), 2014

Meredydd Luff (University of Cambridge), 2012

Demian Rosas Ham (University of Manchester), 2012

PhD Committee: Internal

Erik Tomusk, 2016

Kiran Chandramohan, 2016

Alexander Collins, 2015

Murali Krishna Emani, 2015

Karthik Thucanakkenpalayam Sundararajan, 2013

Oscar Almer, 2012

Damon Fenacci, 2012

Luis Fabricio Goes, 2011

Richard Bennett, 2011

Pedro Diaz, 2011

JOURNAL PUBLICATIONS[2]

AURO	J. Cano, D. R. White, A. Bordallo, C. McCreesh, A. L. Michala, J. Singer, and V. Nagarajan, "Solving the Task Variant Allocation Problem in Distributed Robotics" *Springer Autonomous Robots*, April 2018.
TACO	S. Dublish, V. Nagarajan, and N. Topham, "Cooperative Caching for GPUs," *ACM Transactions on Architecture and Code Optimization*, January 2017.
*TACO**	A.J. McPherson, V. Nagarajan, S. Sarkar and M. Cintra, "Fence placement for legacy data-race-free programs via synchronization read detection," *ACM Transactions on Architecture and Code Optimization*, January 2016.
IJPP	C. Lin, V. Nagarajan, and R. Gupta, "Efficient Sequential Consistency Using Conditional Fences," *International Journal of Parallel Programming*, Vol. 40, No. 1, pages 84-117, special issue of Best Papers from PACT 2010, Feb 2012.
SP&E	V. Nagarajan, D. Jeffrey, R. Gupta, and N. Gupta, "A System for Debugging via Online Tracing and Dynamic Slicing," *Software - Practice and Experience*, 26 pages, published online, July, 2011.
TOPLAS	D. Jeffrey, V. Nagarajan, R. Gupta, and N. Gupta, "Execution Suppression: An Automated Iterative Technique for Locating Memory Bugs," *ACM Transactions on Programming Languages and Systems*, Vol. 32, No. 5. Article No. 17, 36 pages, May 2010.
IJPP	C. Tian, M. Feng, V. Nagarajan, and R. Gupta, "Speculative Parallelization of Sequential Loops on Multicores," *International Journal of Parallel Programming*, Vol. 37, No. 5, pages 508-535, October 2009.

[2]Contributions with an asterisk denote my most significant ones.

SP&E C. Tian, V. Nagarajan, R. Gupta, and S.Tallam "Automated Dynamic Detection of Busy-Wait Synchronizations," *Software - Practice and Experience*, Vol. 39, No. 11, pages 942-972, August 2009.

Trans. HiPEAC V. Nagarajan, R. Gupta, and A.Krishnaswamy, "Compiler-Assisted Memory Encryption for Embedded Processors," *Transactions on High Performance Embedded Architectures and Compilers*, Vol. 2, No. 1, pages 23-44, Springer Verlag, 2009 (Invited Paper – special issue of selected papers from HiPEAC Conference).

SIGOPS V. Nagarajan, and R. Gupta, "Runtime Monitoring on Multicores via OASES," *ACM SIGOPS Operating Systems Review*, special issue on the interaction among the OS, Compilers, and Multicore Processors, Vol. 43, No. 2, pages 15-24, April 2009 (Invited Paper).

CONFERENCE PUBLICATIONS

HPCA S. Dublish, V. Nagarajan, and N. Topham, "Poise : Balancing Thread-Level Parallelism and Memory System Performance in GPUs using Machine Learning," *25th IEEE International Symposium on High-Performance Computer Architecture*, Feb 2019.

*ISCA** N. Oswald, V. Nagarajan, D. Sorin "ProtoGen: Automatically Generating Directory Cache Coherence Protocols from Atomic Specifications," *The 45th International Symposium on Computer Architecture*, June 2018. **(Designated IEEE MICRO 2018 Top Picks Honourable Mention)**

*ISCA** A. Joshi, V. Nagarajan, M. Cintra, and S. Viglas "DHTM: Durable Hardware Transactional Memory ," *The 45th International Symposium on Computer Architecture*, June 2018.

*EuroSys** V. Gavrielatos, A. Katsarakis, A. Joshi, N. Oswald, B. Grot, and V. Nagarajan , "Scale-Out ccNUMA: Exploiting Skew with Strongly Consistent Caching," *The 13th European Conference on Computer Systems*, April 2018.

*ASPLOS** R. Kumar, B. Grot, and V. Nagarajan, "Blasting Through The Front-End Bottleneck with Shotgun," *The 23rd ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, March 2018.

DATE M. Elver, C. J. Banks, P. Jackson, and V. Nagarajan, "VerC3: A Library for Explicit State Synthesis for Concurrent Systems," *Design, Automation and Test in Europe*, to appear, March 2018.

IROS J. Cano, Y. Yang, B. Bodin, V. Nagarajan, and M. O. Boyle, "Automatic Parameter Tuning of Motion Planning Algorithms" *IEEE/RSJInternational Conference on Intelligent Robots and Systems*, 2018.

*FMCAD** C. J. Banks, M. Elver, R. Hoffmann, S. Sarkar, P. Jackson, and V. Nagarajan, "Verification of a lazy cache coherence protocol against a weak memory model," *The 17th Conference on Formal Methods in Computer-Aided Design, Oct 2017*, to appear, Oct. 2017.

ISPASS S. Dublish, V. Nagarajan, and N. Topham, "Evaluating and Mitigating Bandwidth Bottlenecks Across the Memory Hierarchy in GPUs," *IEEE International Symposium on Performance Analysis of Systems and Software*, April 2017.

*HPCA** A. Joshi, V. Nagarajan, S. Viglas and M. Cintra, "ATOM: Atomic Durability in Non-volatile Memory through Hardware Support for Logging ," *The 23rd IEEE Symposium on High Performance Computer Architecture*, Feb. 2017.

*HPCA** R. Kumar, C. Huang, B. Grot, and V. Nagarajan, "Boomerang: a Metadata-Free Architecture for Control Flow Delivery ," *The 23rd IEEE Symposium on High Performance Computer Architecture*, Feb. 2017.

*MICRO**	C. Huang, R. Kumar, M. Elver, B. Grot, and V. Nagarajan "C3D: Mitigating NUMA Effects via Coherent DRAM Caches" *The 49th ACM/IEEE International Symposium on Microarchitecture*, Oct. 2016
*SC**	C. Huang, V. Nagarajan and A. Joshi "DCA: a DRAM-cache-aware DRAM controller" *The IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov. 2016.
IROS	J. Cano, A. Bordallo, V. Nagarajan, S. Ramamoorthy, and S. Vijayakumar "Automatic Configuration of ROS Applications for near-optimal Performance" *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016.
*RSS**	J. Cano, D. White, A. Bordallo, C. McCreesh, P. Prosser, J. Singer and V. Nagarajan, "Task Variant Allocation in Distributed Robotics" *Robotics: Science and Systems*, Ann Arbor, USA, June 2016.
*HPCA**	M. Elver, and V. Nagarajan, "McVerSi: A Test Generation Framework for Fast Memory Consistency Verification in Simulation" *The 22nd Symposium on High Performance Computer Architecture*, Barcelona, Spain, March 2016.
*MICRO**	A. Joshi, V. Nagarajan, S. Viglas and M. Cintra, "Efficient Persist Barriers for Multicores," *IEEE/ACM 48th International Symposium on Microarchitecture*, to appear, Dec. 2015.
*PACT**	M. Elver, and V. Nagarajan, "RC3: Consistency directed Cache Coherence for x86-64 with RC extensions" *The 24th International Conference on Parallel Architectures and Compilation Techniques*, San Francisco, USA, October 2015.
SAFECOMP	G. Stefanakis, V. Nagarajan and M. Cintra, "Understanding the Effects of Data Corruption on Application Behavior Based on Data Charecterestics" *International Conference on Computer Safety, Reliability and Security* , Delft, Netherlands, September 2015.
ICAR	J. Cano, E. Molinos, V. Nagarajan and S. Vijayakumar, "Dynamic process migration in heterogeneous ROS-based environments" *The 17th International Conference on Advanced Robotics* , Istanbul, Turkey, July 2015.
SC	C. Lin, V. Nagarajan, and R. Gupta, "Fence Scoping," *The IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis*, New Orleans, Louisiana, November 2014.
ICCD	C. Huang, and V. Nagarajan, "Increasing Cache Capacity via Critical-words-Only Cache" *The 32nd IEEE International Conference on Computer Design*, Seoul, Korea, October 2014.
*PACT**	C. Huang, and V. Nagarajan, "ATCache: Reducing DRAM cache Latency via a Small SRAM Tag Cache" *The 23rd International Conference on Parallel Architectures and Compilation Techniques*, Edmonton, Canada, August 2014.
*HPCA**	M. Elver, and V. Nagarajan, "TSO-CC: Consistency directed cache coherence for TSO" *The International Symposium on High-Performance Computer Architecture*, Orlando, Florida, February 2014.
*PLDI**	B.Rajaram, V. Nagarajan, S. Sarkar, and M.Elver, "Fast RMWs for TSO: Semantics and Implementation," *ACM SIGPLAN Conference on Programming Language Design and Implementation*, Seattle, Washington, June 2013.
ICS	C.Lin, V. Nagarajan, and R. Gupta, "Address-aware Fences," *27th International Conference on Supercomputing*, Eugene, Oregon, June 2013.

CF B.Rajaram, V. Nagarajan, A.J.McPherson, and M. Cintra, "SuperCoP: A General Correct and Performance-efficient Supervised Memory System," *ACM International Conference on Computing Frontiers*, May 2012.

*ASPLOS** C.Lin, V. Nagarajan, R. Gupta, and B.Rajaram, "Efficient Sequential Consistency via Conflict Ordering," *ACM 17th International Conference on Architectural Support for Programming Languages and Operating Systems*, London, UK, March 2012.

*PACT** C.Lin, V. Nagarajan, and R. Gupta, "Efficient Sequential Consistency Using Conditional Fences," *19th International Conference on Parallel Architectures and Compilation Techniques* , pages 295-306, Vienna, Austria, September 2010. **(Recipient of a Best Paper Award)**

ISMM V. Nagarajan, D.Jeffrey and R. Gupta, "Self-Recovery in Server Programs," *8th International Symposium on Memory Management*, pages 49-58, Dublin, Ireland, June 2009.

*ISCA** V. Nagarajan, and R. Gupta, "ECMon: Exposing Cache events for Monitoring," *ACM/IEEE 36th International Symposium on Computer Architecture*, pages 349-360, Austin, Texas, June 2009.

*VEE** V. Nagarajan, and R. Gupta, "Architectural Support for Shadow Memory in Multiprocessors," *ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, pages 1-10, Washington DC, March 2009.

*MICRO** C. Tian, M. Feng, V. Nagarajan, and R. Gupta, "Copy or Discard Execution Model For Speculative Parallelization On Multicores," *IEEE/ACM 41st International Symposium on Microarchitecture*, pages 330-341, Lake Como, Italy, Nov. 2008.

*ISSTA** C. Tian, V. Nagarajan, R. Gupta, and S. Tallam, "Dynamic Recognition of Synchronization Operations for Improved Data Race Detection," *SIGSOFT International Symposium on Software Testing and Analysis*, pages 143-154, Seattle, July 2008.

ICSM V. Nagarajan, D. Jeffrey, R. Gupta, and N. Gupta, "ONTRAC: A System for Efficient ONline TRACing for Debugging," *International Conference on Software Maintenance*, pages 445-454, Paris, September 2007.

ICSM V. Nagarajan, R. Gupta, X. Zhang, M. Madou, B. De Sutter, and K. De Bosschere, "Matching Control Flow of Program Versions," *International Conference on Software Maintenance*, pages 84-93, Paris, September 2007.

*HiPEAC** V.Nagarajan, R. Gupta, and A.Krishnaswamy, "Compiler-Assisted Memory Encryption for Embedded Processors," *International Conference on High Performance Embedded Architectures and Compilers*, Springer Verlag, LNCS 4367, pages 7-22, Ghent, Belgium, January 2007.

WORKSHOP PUBLICATIONS

NVMW M. Cintra, A. Chatzistergiou, A. Joshi, V. Nagarajan, and S. Viglas, "Architectural Support for Atomic Durability in Non-Volatile Memory," *Non-Volatile Memories Workshop*, San Diego, USA, March 2018. (**Memorable Paper Award Finalist)**

NVMW A. Joshi, V. Nagarajan, S. Viglas, and M. Cintra "DAPPER: a database-inspired approach to persistent memory," *Non-Volatile Memories Workshop*, San Diego, USA, March 2015.

LCPC A. J. McPherson, V. Nagarajan, and M. Cintra, "Static Approximation of MPI Communication Graphs for Optimized Process Placement," *27th International Workshop on Languages and Compilers for Parallel Computing*, Hilsboro, USA, September 2014.

LCPC V. Nagarajan and R. Gupta, "Speculative Optimizations for Parallel Programs on Multicores," *22nd International Workshop on Languages and Compilers for Parallel Computing*, Newark, Delaware, October 2009.

PADTAD	V. Nagarajan and R. Gupta, "Support for Symmetric Shadow Memory in Multiprocessors," *Workshop on Parallel and Distributed Systems: Testing, Analysis, and Debugging* (co-located with ISSTA), 9 pages, Seattle, July 2008.
NSFNGS	R. Gupta, N. Gupta, X. Zhang, D. Jeffrey, V. Nagarajan, S. Tallam and C. Tian, "Scalable Dynamic Information Flow Tracking and its Applications," *NSF Next Generation Software Workshop* (co-located with IPDPS), 5 pages, Florida, April 2008.
STMCS	C. Tian, V. Nagarajan, and R. Gupta, "Synchronization Aware Conflict Resolution for Runtime Monitoring Using Transactional Memory," *Workshop on Software Tools for Multicore Systems* (co-located with CGO), 6 pages, Boston, April 2008.
INTERACT	V. Nagarajan, H-S.Kim, Y.Wu and R. Gupta, "Dynamic Information Flow Tracking on Multicores," *Workshop on Interaction between Compilers and Computer Architectures* (co-located with HPCA), 10 pages, Salt Lake City, Feb. 2008.

PATENT

US Patent 8,321,840: "Software Flow Tracking using Multiple Threads", Inventors: V. Nagarajan, H-S. Kim (Intel), Y. Wu (Intel), and R. Gupta.

TEACHING EXPERIENCE

CS4/MSc PA: Parallel Architectures: 2012/13, 2013/14, 2014/15, 2015/16, 2016/17, 2017/18

INF3 CAR: Computer Architecture: 2011/12, 2012/13 (with Prof. Nigel Topham), 2013/14, 2014/15, 2015/16, 2016/17, 2017/18 (with Dr. Boris Grot)

iSLI : Microprocessors and Microcontrollers 2009/10

MASTERS AND UNDERGRADUATE STUDENTS

Scott Murray, Masters Project, 2014, *Distinction and Prize*.

Vinu Shankar Gopalan, Masters Project, 2012, *Distinction*.

Christian Lalanne, Masters project, 2012, *Distinction*.

Romil Lehakra, Masters Project, 2011

Ross Hamilton, Bachelors Project), 2011, *Distinction*.

PROFESSIONAL ACTIVITIES

Associate Editor: IEEE CAL

General Chair: LCTES 2017

Workshop Co-organizer: WAMS 2018

Finance Chair: PLDI 2014

Registration Chair: PACT 2013

Programme Committee Member: HPCA 2019/2017, ASPLOS 2019 (ERC), CC 2018, SC 2016, ICCD 2015/2014, RTCSA 2016/2015/2014, NVMSA 2017/2016/2015, NAS 2015/2014, WODA 2014, MeAOW 2014, PACT 2018/2014 (ERC), PLDI 2014 (ERC), ICS 2014 (ERC), ASPLOS Doctoral Workshop 2012 (Mentor).

Steering Committee Member: LCTES, Numerical Algorithms and Intelligent Software (NAIS)

Member: RISC-V Memory Model Task group

Member: ACM, IEEE, HiPEAC

Reviewer: NSERC Canada (grant reviewer), EPSRC (grant reviewer), ACM PLDI, ACM TACO, IEEE TC, JPDC, IEEE HPCA, IEEE TSC, ACM CF

ADMINISTRATIVE ACTIVITIES

Masters Project Coordinator, 2016/17

UG4 Personal Tutor, 2014/15, 2015/16, 2016/17

UG3 Course Organiser, 2012/13, 2013/14, 2014/15, 2015/16

PhD Selection Committee, ICSA, 2011-2013

Seminar Organiser, ICSA, 2011-2013