

JUSTIN EMILE GOTTSCHLICH

Intel Corporation, Santa Clara, CA | justin.e.gottschlich@intel.com / justin@nodeka.com | <http://justingottschlich.com>

RESEARCH INTERESTS

I am interested in advancing the field of parallel computing to reduce the challenge of writing correct and efficient parallel programs. I have a specific research emphasis on parallel algorithms, transactional memory, parallel programming language constructs, and the verification, testing, and debugging of parallel software. I am also interested in machine learning.

EDUCATION

Ph.D. in Electrical, Computer, and Energy Engineering University of Colorado at Boulder [Advisor: Jeremy Siek, Co-Advisor: Manish Vachharajani] Dissertation: "Invalidating Transactions: Optimizations, Theory, Guarantees, and Unification"	2011
M.S. in Electrical and Computer Engineering University of Colorado at Boulder [Advisor: Dan Connors] Thesis: "Exploration of Lock-Based Software Transactional Memory"	2007
B.S. in Computer Science Colorado State University	2002

AWARDS

Intel Labs Teamwork Award: Record and Replay Hardware Prototype	2012
Winner: Best Demonstration Award (Intel's Software Professionals Conference)	2011
Inclusion: Marquis Who's Who in Science and Engineering	2011
Winner: Best Presentation Award (IEEE/ACM Symposium on Code Generation and Optimization, CGO)	2010
Winner: Best Presentation Award (Raytheon's Information Systems and Computing Symposium)	2009
Peer-Selection: Technical Honors (Raytheon Company, acceptance rate: ~3.5%)	2008
Peer-Selection: Technical Honors (Raytheon Company, acceptance rate: ~6.0%)	2006

PROFESSIONAL EXPERIENCE

Intel Corporation (Programming Systems Lab, Intel Labs) Research Scientist	2010-Present 2010-Present
University of Colorado-Boulder Adjunct Professor (Department of Electrical, Computer, and Energy Engineering)	2011-Present 2011-Present
Nodeka, LLC. Founder / CEO	1998-Present 1998-Present
Raytheon Company Principal Software Engineer Senior Software Engineer II <i>with Honors</i> Senior Software Engineer II Senior Software Engineer <i>with Honors</i> Senior Software Engineer	2004-2010 2010 2009-2010 2007-2009 2006-2007 2004-2006
Quark Inc. Software Engineer II Software Engineer	1998-2000/2001-2004 2001-2004 1998-2000
KORE Inc. Senior Software Engineer	2000-2001 2000-2001

Software Engineer

2000-2000

GRANTS

-
1. "Simplifying Parallel Programming with Transactional Memory" (PI, \$50,000, Raytheon Company) 2010

PATENTS

-
2. "Concurrent Predicate Expression for Pin-pointing Concurrency Bugs" 2011
Youfeng Wu, **Justin Gottschlich**, Gilles Pokam, Shiliang Hu, Ali-reza Adl-tabatabai, and Cristiano Pereira (patent pending)
1. "Concurrent Predicates" 2011
Justin Gottschlich, Gilles Pokam, Cristiano Pereira, and Jungwoo Ha (patent pending)

PUBLICATIONS**SPECIFICATIONS**

- "Draft Specification of Transactional Language Constructs for C++" 2010-Present
Editors: Ali-Reza Adl-Tabatabai, Tatiana Shpeisman, and **Justin Gottschlich**

Contributors: Ali-Reza Adl-Tabatabai (Intel), Kit Barton (IBM), Hans Boehm (HP), Calin Cascaval (IBM), Steve Clamage (Oracle), Robert Geva (Intel), **Justin Gottschlich** (Intel), Richard Henderson (Red Hat), Victor Luchangco (Oracle), Virendra Marathe (Oracle), Maged Michael (IBM), Mark Moir (Oracle), Ravi Narayanaswamy (Intel), Clark Nelson (Intel), Yang Ni (Intel), Daniel Nussbaum (Oracle), Torvald Riegel (Red Hat), Tatiana Shpeisman (Intel), Raul Silvera (IBM), Xinmin Tian (Intel), Douglas Walls (Oracle), Adam Welc (Intel), Michael Wong (IBM), Peng Wu (IBM)

REFEREED CONFERENCES AND WORKSHOPS

16. "Practical Memory Race Recorder for x86 Multi-core Processors" (21% acceptance, 44/209) MICRO 2011
Gilles Pokam, Cristiano Pereira, Shiliang Hu, Ali-Reza Adl-Tabatabai, **Justin Gottschlich**, Jungwoo Ha, and Youfeng Wu
International Symposium on Microarchitecture (MICRO)
15. "Programming with Concurrent Predicates" (47% acceptance, 313/661) (**Best Demonstration Award**) ISWPC 2011
Justin Gottschlich, Cristiano L. Pereira, Gilles A. Pokam, and Jungwoo Ha
Intel Software Professionals Conference (SWPC)
14. "Optimizing the Concurrent Execution of Locks and Transactions" (37% acceptance, 19/52) LCPC 2011
Justin Gottschlich and JaeWoong Chung
International Workshop on Languages and Compilers for Parallel Computing (LCPC)
13. "Catching Two Birds with One Stone: Isolating Atomicity Violation and Data-Race from Replay and Branch Traces" (47% acceptance, 313/661) ISWPC 2011
Jungwoo Ha, **Justin Gottschlich**, Cristiano L. Pereira, Gilles A. Pokam, and Youfeng Wu
Intel Software Professionals Conference (SWPC)
12. "A New Era of Debugging: Tracing Mechanism for Debugging Multi-core Atom" (Abstract) TechFest 2011
Gilles Pokam, **Justin Gottschlich**, Jungwoo Ha, and Cristiano Pereira
Intel TechFest
11. "Reducing the Integration Complexity of Software Transactional Memory with TBoost.STM" BoostCon 2010
Vicente Botet, **Justin Gottschlich**, and Dwight Winkler
Boost Libraries Conference (BoostCon)
10. "Proving Conflict Serializability for Full Invalidation" WTTM 2010
Justin Gottschlich, Jeremy G. Siek, and Manish Vachharajani
Workshop on the Theory of Transactional Memory (WTTM)
9. "An Efficient Software Transactional Memory Using Commit-Time Invalidation" (41% acceptance, 29/70) (**Best Presentation Award**) CGO 2010
Justin Gottschlich, Manish Vachharajani, and Jeremy G. Siek
IEEE/ACM Symposium on Code Generation and Optimization (CGO)

8. *"An Efficient Lock-Aware Transactional Memory Implementation"* IC00OLPS 2009
Justin Gottschlich, Jeremy G. Siek, Manish Vachharajani, Dwight Winkler, and Daniel A. Connors
 ACM Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (IC00OLPS)
7. *"Toward Simplified Parallel Support in C++"* BoostCon 2009
Justin Gottschlich, Jeremy G. Siek, Paul Rogers, and Manish Vachharajani
 Boost Libraries Conference (BoostCon)
6. *"Shifting the Parallel Programming Paradigm"* (27% acceptance) **(Best Presentation Award)** ISaC 2009
Justin Gottschlich, Mark Holmes, Jeremy G. Siek, and Manish Vachharajani
 Raytheon Information Systems and Computing Symposium
5. *"Lock-Aware Transactional Memory"* (Poster) ASPLOS 2009
Justin Gottschlich, Daniel A. Connors, Dwight Winkler, Jeremy G. Siek, and Manish Vachharajani
 ACM Architectural Support for Programming Languages and Operating Systems (ASPLOS)
4. *"Optimizing Consistency Checking for Memory-Intensive Transactions"* (Brief announcement) PODC 2008
Justin Gottschlich and Daniel A. Connors
 ACM Symposium on Principles of Distributed Computing (PODC)
3. *"C++ Move Semantics for Exception Safety and Optimization in Software Transactional Memory Libraries"* IC00OLPS 2008
Justin Gottschlich, Jeremy G. Siek, and Daniel A. Connors
 ACM Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (IC00OLPS)
2. *"Extending Contention Managers for User-Defined Priority-Based Transactions"* EPHAM 2008
Justin Gottschlich and Daniel A. Connors
 ACM Workshop on Exploiting Parallelism with Transactional Memory and other Hardware Assisted Methods (EPHAM)
1. *"DracoSTM: A Practical C++ Approach to Software Transactional Memory"* (52% acceptance, 11/21) LCSD 2007
Justin Gottschlich and Daniel A. Connors
 ACM International Symposium on Library-Centric Software Design (LCSD)

PRESENTATIONS

13. *"Programming with Concurrent Predicates"* **(Best Demonstration Award)** SWPC 2011
Justin Gottschlich
 Intel's Software Professionals Conference (SWPC)
12. *"Optimizing the Concurrent Execution of Locks and Transactions"* LCPC 2011
Justin Gottschlich
 Languages and Compilers for Parallel Computation (LCPC'11)
11. *"Draft Specification of Transactional Language Constructs for C++"* BoostCon 2011
Justin Gottschlich
 Acknowledgements: Hans Boehm, Mark Moir, Victor Luchangco, Tatiana Shpeisman, and Michael Wong
 Boost Libraries Conference (BoostCon)
10. *"Proving Conflict Serializability for Full Invalidation"* WTTM 2010
Justin Gottschlich, Jeremy G. Siek, and Manish Vachharajani
 Workshop on the Theory of Transactional Memory (WTTM)
9. *"An Efficient Software Transactional Memory Using Commit-Time Invalidation"* **(Best Presentation Award)** CGO 2010
Justin Gottschlich, Manish Vachharajani, and Jeremy G. Siek
 IEEE/ACM Symposium on Code Generation and Optimization (CGO)
8. *"An Efficient Lock-Aware Transactional Memory Implementation"* IC00OLPS 2009
Justin Gottschlich, Jeremy G. Siek, Manish Vachharajani, Dwight Winkler, and Daniel A. Connors
 ACM Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (IC00OLPS)

7. *"Toward Simplified Parallel Support in C++"* BoostCon 2009
Justin Gottschlich, Jeremy G. Siek, Paul Rogers, and Manish Vachharajani
 Boost Libraries Conference (BoostCon)
6. *"Shifting the Parallel Programming Paradigm" (Best Presentation Award)* ISaC 2009
Justin Gottschlich, Mark Holmes, Jeremy G. Siek, and Manish Vachharajani
 Raytheon Information Systems and Computing Symposium
5. *"Lock-Aware Transactional Memory" (Poster)* ASPLOS 2009
Justin Gottschlich, Daniel A. Connors, Dwight Winkler, Jeremy G. Siek, and Manish Vachharajani
 ACM Architectural Support for Programming Languages and Operating Systems (ASPLOS)
4. *"Optimizing Consistency Checking for Memory-Intensive Transactions" (Brief announcement)* PODC 2008
Justin Gottschlich and Daniel A. Connors
 ACM Symposium on Principles of Distributed Computing (PODC)
3. *"C++ Move Semantics for Exception Safety and Optimization in Software Transactional Memory Libraries"* IC00OLPS 2008
Justin Gottschlich, Jeremy G. Siek, and Daniel A. Connors
 ACM Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (IC00OLPS)
2. *"Extending Contention Managers for User-Defined Priority-Based Transactions"* EPHAM 2008
Justin Gottschlich and Daniel A. Connors
 ACM Workshop on Exploiting Parallelism with Transactional Memory and other Hardware Assisted Methods (EPHAM)
1. *"DracoSTM: A Practical C++ Approach to Software Transactional Memory"* LCSD 2007
Justin Gottschlich and Daniel A. Connors
 ACM International Symposium on Library-Centric Software Design (LCSD)

INVITED TALKS

- "Optimizing and Profiling Transactional Memory"* 2010
 Intel Labs, Intel Corporation
- "Profiling Transactional Memory"* 2010
 Sun Labs at Oracle, Oracle Corporation
- "Transactional Memory: Programming and Performance"* 2010
 Center for Computation and Technology, Louisiana State University (LSU)
- "Visualizing Transactional Memory"* 2010
 Center for Computation and Technology, Louisiana State University (LSU)

PROJECTS

- Online Game Development, Owner and CEO of Nodeka, LLC.* 1998-Present
 Sole developer of initial 300,000 lines of C++ code (www.nodeka.com).
 Nodeka, LLC.
- Concurrent Predicates* 2010-Present
 Concurrent predicates (CPs) help programming diagnose and reproduce all types of concurrent violations, such as data races, atomicity violations, order violations, deadlocks, livelocks, and priority inversion. CP complements existing approaches by using a programming language construct that can be used to reason about and reproduce highly complex concurrency bugs. CP can also be used to verify bug fixes are correct.
- PerfViz: 3D Profiler for Hardware-Supported Record and Replay for Multicore* 2011
 Designed and implemented 3D memory record and replay profiler that provides a visualization of a multithreaded execution and the shared memory interleavings between threads. The profiler

includes offline soft real-time display of load and store memory operations and forward and backward playback with rate control.

<i>TMProf: Transactional Memory 3D Profiler with Soft Real-Time Playback</i>	2010
Designed and implemented 3D transactional memory (TM) profiler that allows user to see TM executions in 3D. The profiler also includes transparency, multiple execution overlay, and forward and backward playback with rate control.	
<i>InvalSTM</i>	2010
Designed and implemented a software transactional memory system that is competitive with the state-of-the-art TL2. InvalSTM's primary contribution is to use invalidation for conflict detection, rather than validation, using Bloom filters for efficient space and time complexity for transaction read and write sets, coupled with deferred update and lazy conflict resolution.	
<i>TBoost.STM</i>	2009-Present
TBoost.STM is a software transactional memory library aimed at providing industrial-strength interfaces and stability. TBoost.STM's interfaces approach the simplicity of language-based TM while remaining efficient due to many static-time C++ template type expansions.	
<i>Artificial Neural Networks</i>	2007-2010
Designed and implemented over ten different neural networks in C++ for anomaly characterization and run-time classification. This work is classified, so details are omitted, however, the system correctly classified a highly complex anomaly in real-time in December 2009.	
<i>DracoSTM</i>	2006
Designed and implemented a research-based software transactional memory system that explored design limitations of a C++-based library using C++ parametric polymorphism for type-safety and compile-time optimizations and subtype polymorphism for run-time flexibility.	
<i>QuarkXPress Exception Handling Architecture</i>	2004
Designed and implemented QuarkXPress's exception handling architecture. The design derives from <code>std::exception</code> and provides an integral passing interface to legacy C code.	
<i>QuarkXPress Template Memory Model</i>	2002
Designed and implemented QuarkXPress's memory model that is used as the base memory management system for all QuarkXPress boxes in QuarkXPress 6.0.	
<i>C++ Generic Tree Library (<code>core::tree</code>, <code>core::multitree</code>, <code>core::tree_pair</code>, <code>core::multitree_pair</code>)</i>	2002
Designed and implemented a generic tree container in C++ initially used for Nodeka, LLC. Since publishing two articles about these containers and their generic use on GameDev.net, the tree libraries have been used in over a dozen academic and commercial institutions, such as Brown University, Motorola, Intel, and Quark. After initially limiting the use of the tree library to a small subsection of code, Quark Inc. later requested permission to use the tree libraries throughout the entire QuarkXPress source code. I granted Quark permission for this in 2004. All uses of the tree library have been free of charge. The tree library is now under LGPL, Lesser GNU Public License.	

SERVICE

Program committee member, TRANSACT	2012
External reviewer, Architecture Support for Programming Languages and Operating Systems (ASPLOS)	2012
Editor, Draft Specification for Transactional Memory Constructs in C++	2011
Committee member, Draft Specification for Transactional Memory Constructs in C++	2010-2011
Program committee member, BoostCon	2010-2011
External reviewer, International Conference on Runtime Verification (RV)	2011
External reviewer, International Workshop on Languages and Compilers for Parallel Computing (LCPC)	2011
External reviewer, International Conference on Network and Parallel Computing (NPC)	2011
External reviewer, Symposium on Parallelism in Algorithms and Architectures (SPAA)	2011
External reviewer, European Symposium on Programming (ESOP)	2011
External reviewer, Journal of Parallel and Distributed Computing (JPDC)	2010
Moderator / chair, Transactional Memory Workshop, BoostCon	2010
Moderator / chair, High-Performance Computing Track, Raytheon ISaC Conference	2010

External reviewer, Symposium on Code Generation and Optimization (CGO)	2008
External reviewer, Journal of Parallel and Distributed Computing (JPDC)	2008
External reviewer, Symposium on Computer Architecture (ISCA)	2008

MEMBERSHIPS

Member, ACM
Member, IEEE