JUSTIN EMILE GOTTSCHLICH

Intel Corporation, Santa Clara, CA | justin.gottschlich@gmail.com / justin@nodeka.com | http://justingottschlich.com

BRIEF BIO

I am principally interested in machine learning applied to several domains, such as time-series neural networks, anomaly detection systems, and the fusion of programming and machine learning. I have over 20 peer-reviewed technical publications, seven issued patents, and several dozen patents pending. I have been the recipient of several best presentation awards, chaired several workshops, was director of engineering at a top mobile gaming company, and am the founder and CEO of Nodeka, an online gaming company.

EDUCATION

Ph.D. in Electrical, Computer, and Energy Engineering	**2010**
University of Colorado at Boulder [Advisor: Jeremy Siek, Co-Advisor: Manish Vachharajani]	
M.S. in Electrical and Computer Engineering	**2007**
University of Colorado at Boulder [Advisor: Dan Connors]	
B.S. in Computer Science	**2002**
Colorado State University	

LEADERSHIP

Project Lead, Anomalous Dataset Generation (Intel & Stanford)	*2016-2019*
Deputy Technical Lead, Intel & NSF CAPA project ($6M budget)	*2016-2020*
Technical Lead, Anomaly Projects at Intel Labs	2016-2017
Program Chair for MAPL 2017 (1st Workshop on Machine Learning and Programming Languages)	2016-2017
Founding Member of MAPL 2017 (Workshop on Machine Learning and Programming Languages)	2016-present
Director of Engineering, Machine Zone (supervised 50+ engineers)	2015-2016
Adjunct Professor at University of Colorado-Boulder (taught Neural Network graduate class)	2011-present
General Chair for TRANSACT 2015 (10th ACM SIGPLAN Workshop on Transactional Computing)	2014-2015
Steering Committee Member for TRANSACT (Workshop on Transactional Computing)	2015-present
Voting Member of Intel's Intellectual Property Council (Systems Software)	2014-present
Program Chair for TRANSACT 2014 (9th ACM SIGPLAN Workshop on Transactional Computing)	2013-2014
Voting Member of Intel's Intellectual Property Council (Software and Internetworking Technologies)	2013-2014
Vice-Chair of Standard C++ Study Group 5 (SG5: Transactional Memory)	2012-2014
Application Track Chair for TRANSACT 2013 (8th ACM SIGPLAN Workshop on Transactional Computing)	2012-2013
Chair and *Editor* of the "Draft Specification of Transactional Constructs for C++"	2011-2012
Founder and CEO of Nodeka, LLC (wrote ~500k LoC C++)	1999-present

PROJECTS

Anomaly detection and management projects (Intel), Lead Research (supervised 10+ PhD researchers)	2016-present
Transactional HTM/HyTM Debugging (Intel), Lead Researcher	2013-Present
Hybrid Transactional Memory Research (Intel), Lead Researcher	2012-2014
Concurrent Predicates (Intel), Lead Researcher	2011-2013
Memory Race Recorder (Intel), Researcher	2010-Present
Transactional Memory for Standard C++ (Intel), Co-lead Researcher	2010-Present
Transactional Memory Profiling / Visualization (Intel/CU-Boulder), Lead Researcher	2010-Present
Artificial Intelligence for Anomalous Events (Raytheon), Lead Designer and Developer	2007-2010
QuarkXPress Memory Management System (Quark), Lead Developer	2002-2004
QuarkXPress Exceptional Handling System (Quark), Lead Developer	2002-2004
Online Gaming (Nodeka), Lead Designer and Developer (400,000+ lines of C++ code)	1998-Present

AWARDS

Intel Labs Research Velocity Challenge Winner: Anomaly Detection for Data Centers Using Machine Learning	2016
Intel Corporation's High 5 (awarded three times)	2013
Intel Labs Teamwork Award: Record and Replay Hardware Prototype	2012
Winner: Best Demonstration Award (Intel's Software Professionals Conference)	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 (Parallel Computing Lab, Intel Labs) **2016-2017**
Senior Staff Research Scientist 2016-2017

Machine Zone **2015-2016**
Director of Engineer (supervised 50+ engineers) 2015-2016

Intel Corporation (Programming Systems Lab, Intel Labs) **2010-2015**
Staff Research Scientist 2013-2015
Research Scientist 2010-2012

University of Colorado-Boulder **2011-Present**
Adjunct Professor (Department of Electrical, Computer, and Energy Engineering) 2011-Present

Nodeka, LLC. **1998-Present**
Founder / CEO 1998-Present

Raytheon Company **2004-2010**
Principal Software Engineer 2010-2010
Senior Software Engineer II *with Honors* 2009-2010
Senior Software Engineer II 2007-2009
Senior Software Engineer *with Honors* 2006-2007
Senior Software Engineer 2004-2006

Quark Inc. **1998-2000/2001-2004**
Software Engineer II 2001-2004
Software Engineer 1998-2000

KORE Inc. **2000-2001**
Senior Software Engineer 2000-2001
Software Engineer 2000-2000

STUDENTS

PhD co-advisor: Irina Calciu, Brown University (advisor: Maurice Herlihy) 2012-2015
PhD committee member: Wenjia Ruan, Lehigh University (advisor: Mike Spear) 2014-2015

PUBLICATIONS

SPECIFICATIONS

Draft Specification of Transactional Language Constructs for C++ (v.1.1) 2010-2012
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 CONFERENCE AND WORKSHOP PAPERS

24. TSXProf: Profiling Hardware Transactions (21% acceptance, 38/179) PACT 2015
Yujie Liu, Justin Gottschlich, Gilles Pokam, and Michael Spear

23. *Towards Transactional Memory for OpenMP* IWOMP 2014
Michael Wong, Eduard Ayguade, **Justin Gottschlich**, Victor Luchangco, Bronis R. de Supinski, and Barna Bilhari

International Workshop on OpenMP (IWOMP)

22. *Invyswell: A Hybrid Transactional Memory for Haswell's Restricted Transactional Memory* PACT 2014
(25% acceptance, 37/149)
Irina Calciu, **Justin Gottschlich**, Tatiana Shpeisman, Gilles Pokam, and Maurice Herlihy
International Conference on Parallel Architectures and Compilation Techniques (PACT)

21. *Concurrent Predicates: A Debugging Technique for Every Parallel Programmer* (17% acceptance, 36/208) PACT 2013
Justin Gottschlich, Gilles Pokam, Cristiano Pereira, and Youfeng Wu
International Conference on Parallel Architectures and Compilation Techniques (PACT)

20. *But How Do We Really Debug Transactional Memory Programs?* (48% acceptance, 14/30) HotPar 2013
Justin Gottschlich, Rob Knauerhase, and Gilles Pokam
USENIX Workshop on Hot Topics in Parallelism (HotPar)

19. *Using Elimination and Delegation to Implement a Scalable NUMA-Friendly Stack* (48% acceptance, 14/30) HotPar 2013
Irina Calciu, **Justin Gottschlich**, and Maurice Herlihy
USENIX Workshop on Hot Topics in Parallelism (HotPar)

18. *QuickRec: Prototyping an Intel Architecture Extension for Record and Replay of Multithreaded Programs* ISCA 2013
(19% acceptance, 56/288)
Gilles Pokam, Klaus Danne, Cristiano Pereira, Rolf Kassa, Tim Kranich, Shiliang Hu, **Justin Gottschlich**,
Nima Honarmand, Nathan Dautenhahn, Sam King, and Josep Torrellas
International Symposium on Computer Architecture (ISCA)

17. *Generic Programming Needs Transactional Memory* TRANSACT 2013
Justin Gottschlich and Hans-J. Boehm
ACM SIGPLAN Workshop on Transactional Computing (TRANSACT)

16. *Visualizing Transactional Memory* (19% acceptance, 39/207) PACT 2012
Justin Gottschlich, Maurice Herlihy, Gilles Pokam, and Jeremy Siek
International Conference on Parallel Architectures and Compilation Techniques (PACT)

15. *Concurrent Predicates: Finding and Fixing the Root Cause of Concurrency Violations* HotPar 2012
Justin Gottschlich, Gilles Pokam, and Cristiano Pereira
USENIX Workshop on Hot Topics in Parallelism (HotPar)

14. *CoreRacer: A Practical Memory Race Recorder for x86 TSO 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)

13. *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)

12. *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)

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* ICOOOLPS 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 (ICOOOLPS)

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* — ICOOOLPS 2008
Justin Gottschlich, Jeremy G. Siek, and Daniel A. Connors
ACM Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages, Programs and Systems (ICOOOLPS)

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)

GRANTS

2. Discovering hidden structure in high dimensional data for anomaly detection (Intel's URO Sponsorship) — 2017-2020
(co-authored with Prof. Stefano Ermon, Stanford University, $300,000)
1. *Simplifying Parallel Programming with Transactional Memory* (PI, $50,000, Raytheon Company) — 2010

PATENTS ISSUED

1 9,501,340 T Mechanism for facilitating dynamic and efficient management of instruction atomicity violations in software programs at computing systems
2 9,361,152 T Transactional memory management techniques
3 9,317,297 T Replay execution of instructions in thread chunks in the chunk order recorded during previous execution
4 9,311,143 T Methods and systems to identify and reproduce concurrency violations in multi-threaded programs
5 9,135,139 T Methods and systems to identify and reproduce concurrency violations in multi-threaded programs using expressions
6 9,128,781 T Processor with memory race recorder to record thread interleavings in multi-threaded software
7 9,117,021 T Methods and apparatus to manage concurrent predicate expressions

PRESENTATIONS

18. Using Machine Learning to Avoid the Unwanted — Intel's HPC DevCon 2016
Justin Gottschlich
Intel's High Performance Computer Developers Conference

17. A Debugging Technique for Every Parallel Programmer — PACT 2013

Justin Gottschlich
International Conference on Parallel Architectures and Compilation Techniques (PACT)

16. But How Do We *Really* Debug Transactional Memory Programs? HotPar 2013
Justin Gottschlich
USEXNIX Workshop on Hot Topics in Parallelism (HotPar)

15. Generic Programming Needs Transactional Memory TRANSACT 2013
Justin Gottschlich
ACM SIGPLAN Workshop on Transactional Computing (TRANSACT)

14. *Concurrent Predicates: Finding and Fixing the Root Cause of Concurrency Violations* HotPar 2012
Justin Gottschlich
USENIX Workshop on Hot Topics in Parallelism (HotPar)

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
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
IEEE/ACM Symposium on Code Generation and Optimization (CGO)

8. *An Efficient Lock-Aware Transactional Memory Implementation* ICOOOLPS 2009
Justin Gottschlich
ACM Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages,
Programs and Systems (ICOOOLPS)

7. *Toward Simplified Parallel Support in C++* BoostCon 2009
Justin Gottschlich
Boost Libraries Conference (BoostCon)

6. *Shifting the Parallel Programming Paradigm* ***(Best Presentation Award)*** ISaC 2009
Justin Gottschlich
Raytheon Information Systems and Computing Symposium

5. *Lock-Aware Transactional Memory* (Poster) ASPLOS 2009
Justin Gottschlich
ACM Architectural Support for Programming Languages and Operating Systems (ASPLOS)

4. *Optimizing Consistency Checking for Memory-Intensive Transactions* (Brief announcement) PODC 2008
Justin Gottschlich
ACM Symposium on Principles of Distributed Computing (PODC)

3. *C++ Move Semantics for Exception Safety and Optimization in Software Transactional Memory Libraries* ICOOOLPS 2008
Justin Gottschlich
ACM Workshop on Implementation, Compilation, Optimization of Object-Oriented Languages,
Programs and Systems (ICOOOLPS)

2. *Extending Contention Managers for User-Defined Priority-Based Transactions* EPHAM 2008

Justin Gottschlich
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
ACM International Symposium on Library-Centric Software Design (LCSD)

INVITED TALKS

Concurrent Predicates: A Debugging Technique for Every Parallel Programmer 2014
Lockheed Martin

Concurrent Predicates: A Debugging Technique for Every Parallel Programmer 2013
IBM T.J. Watson

Concurrent Predicates: A Debugging Technique for Every Parallel Programmer 2013
Brown University

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 400,000+ lines of C++ code (www.nodeka.com). Created artificial intelligence scripting language. Developed large scale software system that is highly robust (30+ day uptimes with hundreds of concurrent users and hundreds of thousands of artificial intelligent agents).

Transactional Debugging (TDB) – Lead Researcher 2013-Present
TDB aims to improve the debugging experience of transactional memory programs by providing hardware and software based extensions in the recording and replaying of transactional programs that use HTM, STM, or HyTM. An important characteristic of our system is that it is based in hardware, which incurs negligible overhead, thereby ensuring the original contention signature of the program remains intact.

Hybrid Transactional Memory – Managing Researcher 2013-2014
Assisted in the design and management of implementation of a hybrid transactional memory (HyTM) system. Oversaw the design and implementation of Invyswell (PACT '14), which uses a combination of InvalSTM and Intel's Haswell RTM, to provide an efficient HyTM system for various types of transactions.

Memory Race Recorder – Researcher 2010-Present
The memory race recorder (MRR) project aims to use hardware extensions to capture shared memory interleavings that enable programmers to deterministically reproduce any recorded execution. The MRR hardware incurs negligible overhead, by using chunks to record groupings of non-interleaved instructions, thereby ensuring the original contention signature of a nondeterministic multithreaded execution is captured without perturbation. See our ISCA'12 paper for more details.

Concurrent Predicates – Lead Researcher 2010-2012
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. See our HotPar'12 and PACT'13 papers for more details.

PerfViz: 3D Profiler for Hardware-Supported Record and Replay for Multicore – Lead Researcher 2011-2012
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.

TMProf: Transactional Memory 3D Profiler with Soft Real-Time Playback 2010-2012
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. See our PACT'12 paper for more details.

InvalSTM 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. See our CGO'10 paper for more details.

TBoost.STM 2009-2010
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.

Artificial Neural Networks 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.

DracoSTM 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.

QuarkXPress Exception Handling Architecture 2004
Designed and implemented QuarkXPress's exception handling architecture. The design derives from std::exception and provides an integral passing interface to legacy C code.

QuarkXPress Template Memory Model 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.

C++ Generic Tree Library (core::tree, core::multitree, core::tree_pair, core::multitree_pair) 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

NSF Panel member	2016
Reviewer, High-Performance Computer Architecture (HPCA)	2014
Reviewer, Journal of the ACM (JACM)	2014
Steering committee, TRANSACT	2014-present
General chair, TRANSACT	2014
Program chair, TRANSACT	2013
External reviewer, International Symposium on Distributed Computing (DISC)	2013
Town hall moderator: Language-Level Standards for Transactional Memory (TRANSACT)	2013
External reviewer, Architecture Support for Programming Languages and Operating Systems (ASPLOS)	2013
Application track chair, TRANSACT	2013

Vice-chair, Standard C++'s Study Group (SG5: Transactional Memory)	2012
Chair, Draft Specification for Transactional Memory Constructs in C++	2012
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++	2011-2012
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