

# Chen Tian

---

Samsung Information Systems America Inc.  
Computer Science Laboratory  
75 W Plumeria Drive  
San Jose, CA 92134

E-mail: [chen.tian@samsung.com](mailto:chen.tian@samsung.com)  
Cell: 760-842-2837  
Tel.: 408-544-2701

---

## BRIEF BIOGRAPHY

---

Dr. Chen Tian is a Senior Systems Research Engineer at Samsung Information System America Inc., San Jose, California. His research interests span the areas of software engineering, compilers, runtime systems, programming languages, operating systems and computer architectures. Particularly, he has been working on the following topics : compiler, run-time system and operating system technologies in resource constrained manycore environments; compiling for speculative parallel execution; dynamic analysis for software reliability including debugging and failure-avoidance; architectural support for high performance computing.

Many research papers that Dr. Tian authored or co-authored have been published in top-tier conferences and journals including Conference on Programming Language Design and Implementation (PLDI), International Symposium on Microarchitecture (MICRO), ACM Symposium on Principles and Practice of Parallel Programming (PPoPP), International Symposium on Software Testing and Analysis (ISSTA), and International Journal of Parallel Programming (IJPP). He also served as an external reviewer for many conferences and journals including PLDI, Transactions on Computers (TC) and Parallel Computing (Parco).

Dr. Chen Tian received his Ph.D. degree on Computer Science at the University of California at Riverside in June 2010. Before that he obtained his M.S. degree on Computer Science at the University of Arizona in 2007. He also obtained his M.S. degree on Mathematics at Beijing Jiaotong University in 2005.

## EDUCATION

---

<b>Ph.D. in Computer Science</b> The University of California at Riverside Thesis title: Speculative Parallelization on Multicore Processors	June 2010
<b>M.S. in Computer Science</b> The University of Arizona	May 2007
<b>M.S. in Mathematics</b> Beijing Jiaotong University	May 2005
<b>B.S. in Mathematics</b> Beijing Jiaotong University	May 2002

## RESEARCH EXPERIENCE

---

**Sr. Research Engineer**      **Samsung Information Systems America Inc.**      **July 2010 - Present**

### *Systems Research for Constrained Multicore/Manycore Environments*

We are currently developing novel compiler, run-time and operation system technologies to support highly-efficient scaling in resource constrained multicore and manycore environments. A key emphasis of our work is the ability to support dynamic and adaptive scaling so that changes in resource availability (e.g., through different power-modes) and application load can be seamlessly managed from a Quality-of-Service (QoS) perspective. Our new solution also provides large-scale distributed computation coordination, live task migration, adaptive scheduling and intelligent runtime system for energy and performance.

**Research Assistant**      **University of Arizona and UC Riverside**      **May 2006 - July 2010**

### *Compiling for Speculative Parallel Execution*

The advent of multicores presents a promising opportunity for speeding up sequential programs via profile-based *speculative parallelization* of these programs. We present a novel solution for efficiently supporting software speculation on multicore processors. We propose the Copy or Discard (CorD) execution model in which the state of speculative parallel threads is maintained separately from the non-speculative computation state. If speculation is successful, the results from speculative state are committed to non-speculative state. Otherwise, speculatively computed results of a thread are discarded and the computation is performed again. Our experiments show that the combination of CorD and our speculative parallelization algorithm achieves speedups ranging from 2.6 to 7.9.

We also extend our CorD model to parallelize the applications using dynamic data structures which pose new challenges due to the large number of nodes and internal pointers. The speedup we obtained ranges from 1.7 to 4.3.

High misspeculation rate is one of the main factors that limit the parallelism benefit for some programs. To tackle this problem, we present a multiple value prediction based speculations, which can speedup the loops having strong cross-iteration dependences. The experiment show that we achieve 1.6x speedup on average.

To improve the performance of CorD, we also develop an approach for incremental recovery in which, instead of discarding all of the results and reexecuting the speculative computation in its entirety, the computation is restarted from the earliest point at which a misspeculation causing value is read. Our experiments show that with inputs that result in misspeculation rates of around 40% and 80%, applying incremental recovery technique results in 1.2x-3.3x and 2.0x-6.6x speedups respectively over the original discard-all recovery scheme.

### *Software Debugging, Failure-Avoidance*

In multi-threaded programs, data race errors are very common and hard to detect. We observe that prior race detection works rely on one important assumption: the debuggers are aware of all the synchronization operations that take place during a program run. Unfortunately, this assumption is not true when synchronizations are implemented in the user code. To address this problem, we present a dynamic *synchronization detection* technique that can effectively detect synchronizations and thus improve the accuracy of prior race detectors.

For programs that are multi-threaded and long-running, debugging them using trace based analysis is a very challenging problem due to the non-determinism and the exorbitant tracing overhead. We develop a novel scheme that using *logging/replay* and *execution reduction* techniques to address the challenges. Experiments show that our solution can effectively detect both memory type bugs and data races while it reduces the tracing time by three orders of magnitude and the trace sizes by three to five orders of magnitude.

A large number of failures that occur in today's software, including those causing system crashes or producing wrong visible outputs, are due to the execution environment. These failures are often fixable by safe *execution perturbations* such as changing thread scheduling, padding memory allocations and dropping user requests. We present an online framework to capture and recover from failures by applying the perturbations and prevent them from occurring in the future. Our experiments show that our solution is very effective and entailed a very low overhead (2%-19%) for the normal execution.

## **PUBLICATIONS**

---

- PPoPP      **C. Tian**, C. Lin, M. Feng and R. Gupta, "Enhanced Speculative Parallelization Via Incremental Recovery", *16th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, San Antonio, 2011. Acceptance Rate: **15% (26/165)**.
- SFMA      D. Waddington, **C. Tian** and KC Sivaramakrishnan "Scalable Lightweight Task Management for MIMD Processor", *Systems for Future Multicore Architectures*, Salzburg, Austria, 2011.

PLDI	<b>C. Tian</b> , M. Feng and R. Gupta, "Supporting Speculative Parallelization In The Presence Of Dynamic Data Structures", <i>ACM SIGPLAN 2010 Conference on Programming Language Design and Implementation</i> , Toronto, Canada, 2010. Acceptance Rate: <b>20% (41/204)</b> .
SP&E	D. Jeffrey, Y. Wang, <b>C. Tian</b> and R. Gupta, " Isolating Bugs in Multithreaded Programs Using Execution Suppression", <i>Software: Practice and Experience</i> , accepted October 2010.
IJPP	<b>C. Tian</b> , M. Feng, V. Nagarajan and R. Gupta, "Speculative Parallelization of Sequential Loops On Multicores", <i>International Journal of Parallel Programming</i> , Volume 37, Issue 5, Page 508, 2009.
SP&E	<b>C. Tian</b> , V. Nagarajan, R. Gupta and S. Tallam, "Automated dynamic detection of busy-wait synchronizations", <i>Software: Practice and Experience</i> , Volume 39, Issue 11, Page 947, 2009.
ISMM	<b>C. Tian</b> , M. Feng and R. Gupta, "Speculative Parallelization Using State Separation and Multiple Value Prediction", <i>International Symposium on Memory Management</i> , Toronto, Canada, 2010. Acceptance Rate: <b>43% (13/30)</b> .
MICRO	<b>C. Tian</b> , M. Feng, V. Nagarajan and R. Gupta, "Copy Or Discard Execution Model For Speculative Parallelization On Multicores", <i>IEEE/ACM 41th International Symposium on Microarchitecture</i> , Lake Como, Italy, 2008. Acceptance Rate: <b>19% (40/210)</b> .
ICSM	S. Tallam, <b>C. Tian</b> , and R. Gupta, "Dynamic Slicing of Multithreaded Programs for Race Detection", <i>International Conference on Software Maintenance.</i> , Beijing, China, 2008. Acceptance Rate : <b>25 % (40/156)</b> .
ISSTA	<b>C. Tian</b> , V. Nagarajan, R. Gupta, and S. Tallam, "Dynamic Recognition of Synchronization Operations for Improved Data Race Detection", <i>International Symposium on Software Testing and Analysis.</i> , Seattle, 2008. Acceptance Rate : <b>26% (26/100)</b> .
COMPSAC	S. Tallam, <b>C. Tian</b> , X. Zhang, and R. Gupta, "Avoiding Program Failures through Safe Execution Perturbations", <i>IEEE Computer Software and Applications Conference.</i> , Turku, Finland, 2008. Acceptance Rate : <b>20%</b> .
STMCS	<b>C. Tian</b> , V. Nagarajan and R. Gupta, "Synchronization Aware Conflict Resolution for Runtime Monitoring Using Transactional Memory", <i>Workshop on Software Tools for Multicore Systems</i> , Boston, 2008.
NSFNGS	R. Gupta, N. Gupta, X. Zhang, D. Jeffrey, V. Nagarajan, S. Tallam and <b>C. Tian</b> "Scalable Dynamic Information Flow Tracking and its Applications", <i>NSF Next Generation Software Workshop</i> , Florida, 2008.
ISSTA	S. Tallam, <b>C. Tian</b> , X. Zhang, and R. Gupta, "Enabling Tracing of Long-Running Multithreaded Programs via Dynamic Execution Reduction", <i>International Symposium on Software Testing and Analysis</i> , London, UK, 2007. Acceptance Rate : <b>22% (22/101)</b> .
SIGPLAN	D. Waddington, <b>C. Tian</b> and KC Sivaramakrishnan "Scalable Lightweight Task Management for MIMD Processor", <i>SIGPLAN Notices 2011</i> , submitted.
MICRO	M. Feng, <b>C. Tian</b> , C. Lin and R. Gupta, "Enhancing LRU Replacement via Phantom Associativity", <i>IEEE MICRO</i> , submitted.

#### BOOK CHAPTER

---

- Chapter "Software Based Speculative Parallelization For Multicore/Manycore Architecture" has been accepted in book *Programming Multi-core and Many-core Computing Systems*, to be published in 2011.

#### PATENTS

---

- NUMA Aware System Task Management, US Patent Application #13/077,612, 2011.
- Adaptive Queuing Methodology For System Management, US Patent Application #13/077,567, 2011

#### PRESENTATIONS

---

- "Enhanced Speculative Parallelization Via Incremental Recovery", *16th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, San Antonio, 2011.

- “Supporting Speculative Parallelization In The Presence Of Dynamic Data Structures”, *ACM SIG-PLAN 2010 Conference on Programming Language Design and Implementation*, Toronto, Canada, 2010.
- “Speculative Parallelization Using State Separation and Multiple Value Prediction”, *International Symposium on Memory Management*, Toronto, Canada, 2010.
- “Copy Or Discard Execution Model For Speculative Parallelization On Multicores”, *IEEE/ACM 41th International Symposium on Microarchitecture*, Lake Como, Italy, 2008.
- “Dynamic Slicing of Multithreaded Programs for Race Detection”, *International Conference on Software Maintenance.*, Beijing, China, 2008.
- “Dynamic Recognition of Synchronization Operations for Improved Data Race Detection”, *International Symposium on Software Testing and Analysis.*, Seattle, 2008.
- “Synchronization Aware Conflict Resolution for Runtime Monitoring Using Transactional Memory”, *Workshop on Software Tools for Multicore Systems*, Boston, 2008.

## PROFESSIONAL ACTIVITIES

---

- Member of Association for Computing Machinery (**ACM**).
- Member of Institute of Electrical and Electronics Engineers (**IEEE**).
- Reviewed papers for conferences, journals and books:
  - IEEE Computer Architecture Letters*, 2011
  - IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2011
  - Parallel Computing* 2011
  - Wiley book *Programming Multi-core and Many-core Computing Systems*, 2011
  - Conference on Programming Language Design and Implementation* 2011
  - IEEE Transactions on Computers* 2011
  - Science of Computer Programming* 2011
  - Conference on Programming Language Design and Implementation* 2010
  - IEEE Transactions on Computers* 2010
  - ChinaCom* 2007
  - Parallel Computing* 2007
  - Parallel Computing* 2006