



# ATLAS: A Chip-Multiprocessor with Transactional Memory Support

Njuguna Njoroge, Jared Casper, **Sewook Wee**, Yuriy Teslyar,  
Daxia Ge, Christos Kozyrakis, and Kunle Olukotun

Transactional Coherence and Consistency (TCC) Group

Computer Systems Laboratory

Stanford University

<http://tcc.stanford.edu>

# ATLAS is...

- **8-way Chip-Multiprocessor (CMP) System with Transactional Memory Support**
- **Full System Prototype Implementation on the Berkeley Emulation Engine 2 (BEE2) board with 5 FPGAs**
- **Fast and Practical Software Development Platform**
  - **100x Faster than the equivalent Software Simulator**
  - **Full Operating System Support**

# Outline

- **Introduction**
- **Transactional Memory Overview**
- **ATLAS System Implementation**
- **Evaluation**
- **Conclusion**

# Introduction

- **Chip Multi-Processor (CMP) is NOW**
    - **Increased Transistor budgets**
- ▼
- Scalable Performance**  
w/o power and complexity challenge
- **Diminishing return from single-core chips**
  - **Most of processor vendors are toward CMPs**



QuickTime™ and a  
TIFF (Uncompressed) decompressor  
are needed to see this picture.



QuickTime™ and a  
TIFF (LZW) decompressor  
are needed to see this picture.



# How to program the CMP?

- **Conventional Parallel Programming is Difficult**
  - Fine-grain locking for the high performance
  - Error-prone (Deadlock / Livelock)
- **Transactional Memory (TM) makes it easier**
  - Program with large atomic regions
  - Keep the performance of fine-grained locking
- **TM has been studied actively from academia & industry**
  
- **One Missing Piece:**
  - Nobody made a real system with TM support
  - Fast platform to develop applications

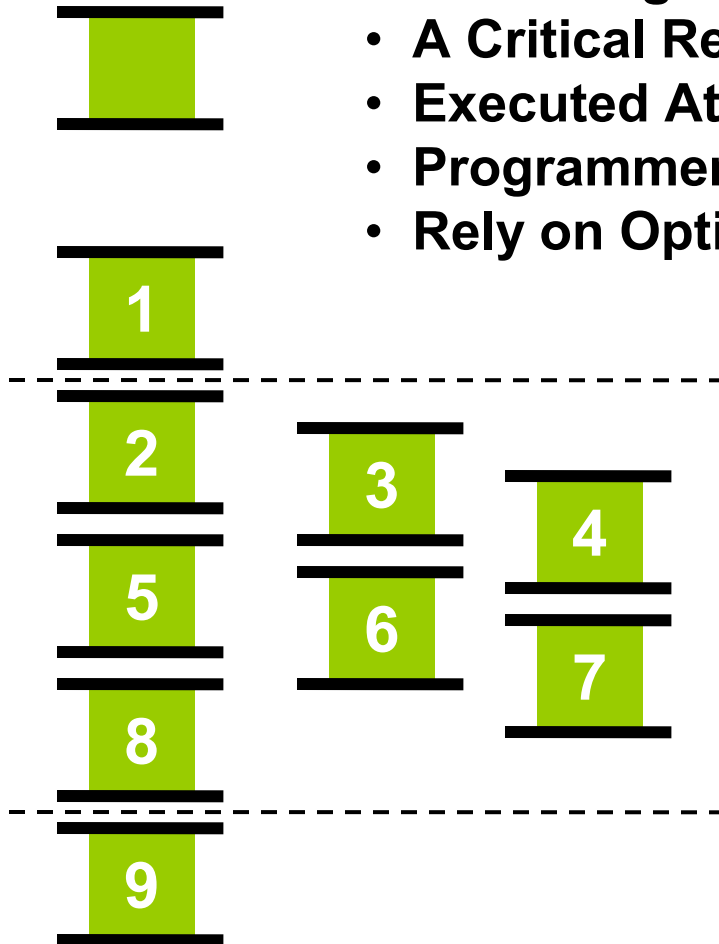
# ATLAS's Contributions

- **The First Hardware Prototype**
  - 8-way CMP with TM support (TCC Protocol)
  - Full system support powered by one separate service processor running Linux OS
- **The First Evaluation of TM on the real system**
  - TM application scales well on ATLAS as promised
- **Fast Software Development Platform**
  - Runs on 100MHz Real Hardware
  - Runtime Performance Profiler
  - Guided Performance Tuning

# Transactional Memory Overview

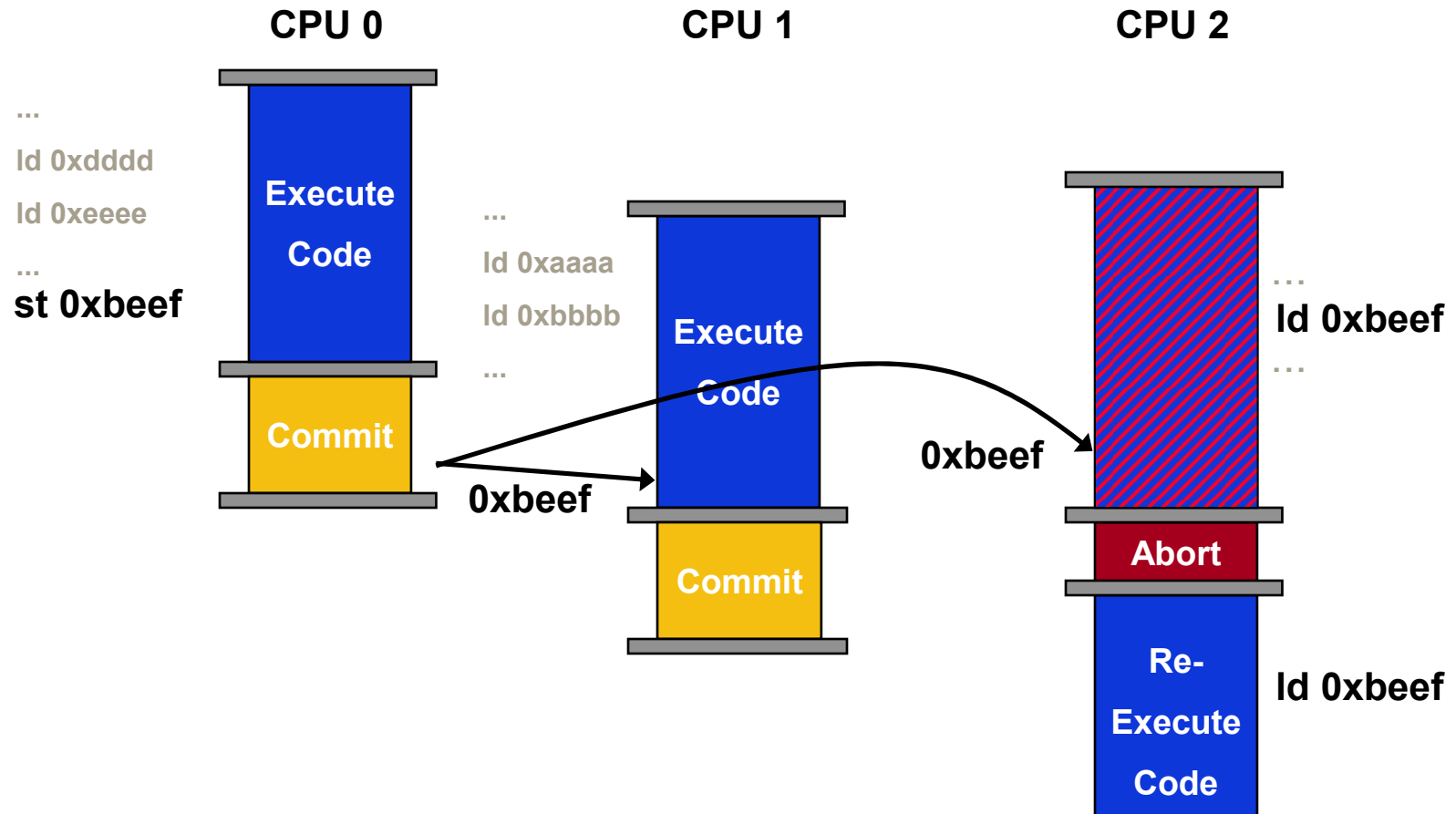
## Transaction:

- A building block of a program
- A Critical Region
- Executed Atomically and Isolated
- Programmer wraps it with TM API
- Rely on Optimistic Concurrency



**Optimistic Concurrent Execution**  
assumes that data conflict happens rarely in the runtime

# TCC Execution Model





# ATLAS Implementation

## ■ Requirements

- Read Set & Write Set Buffer
- Conflict Detection
- Transaction Checkpoint to Rollback

## ■ ATLAS Implementation

- Modified L1 Data Cache for Read Set & Write Set Buffer
- Modified L1 Snooping Hardware for Conflict Detection
- Special Memory for the Register Checkpoint to Rollback

## ■ Issue

- PowerPC 405 hardcore processor on Xilinx Virtex 2 Pro FPGA
- Fast, stable, and abundant software support.
- Disabled internal D-Cache → 13 cycles cache hit latency.

# L1 Cache with TCC support

Transactionally Read Bits:

**Id 0xdeadbeef**

Transactionally Written Bits:

**st 0xcafebabe**

Commit:

Read pointers from Store Address FIFO, flush addresses with W bits set

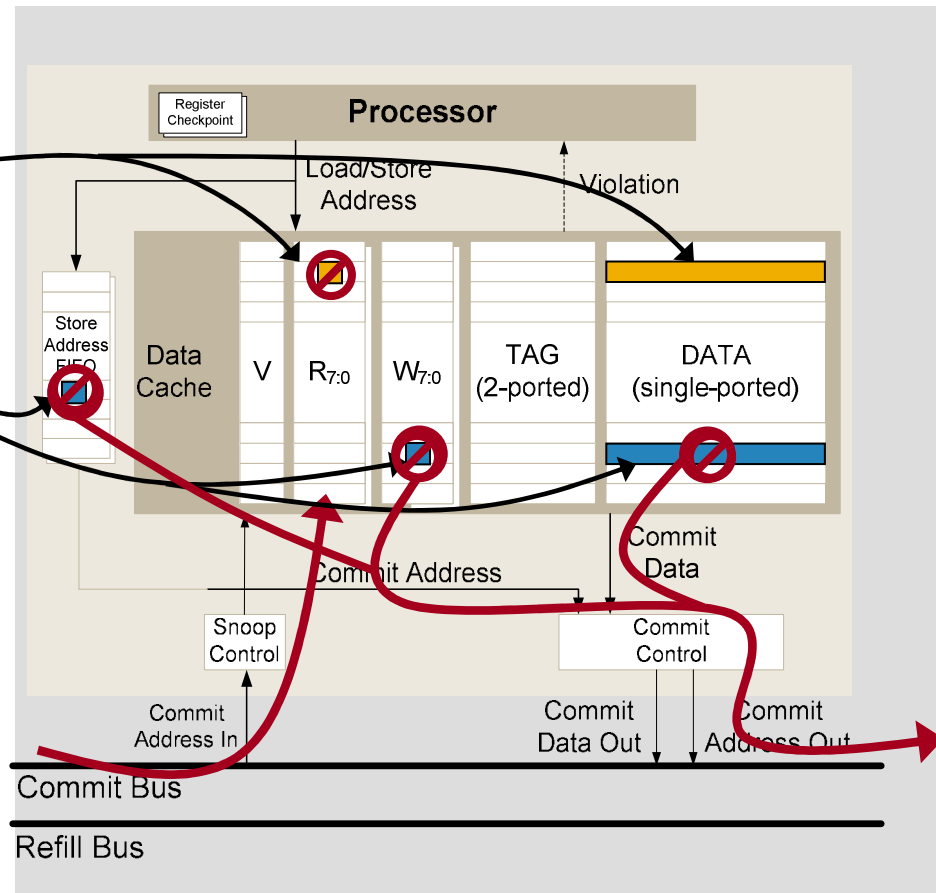
Violation Detection:

Compare incoming address to R bits

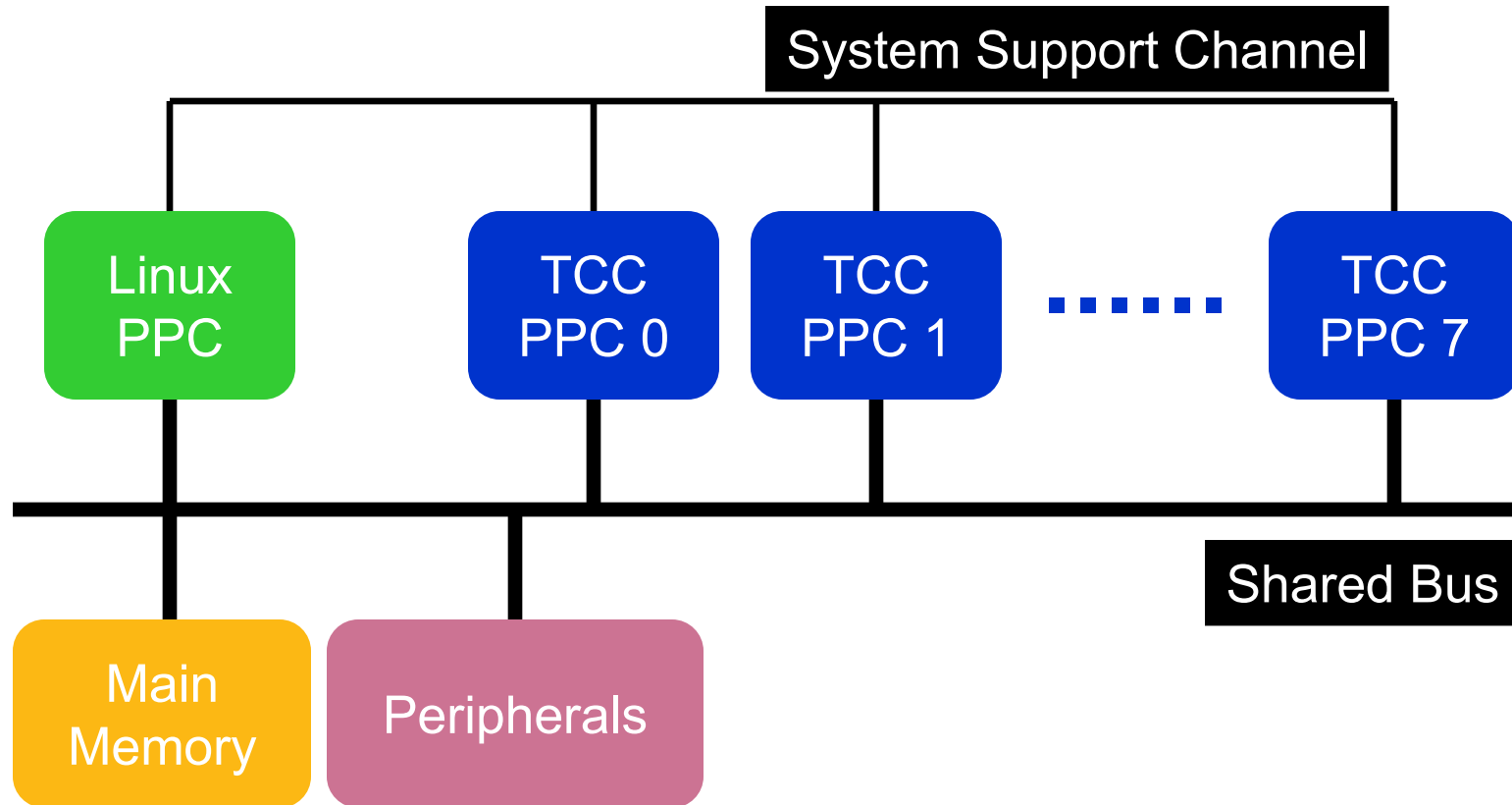
Overflow of

Transactional buffer :

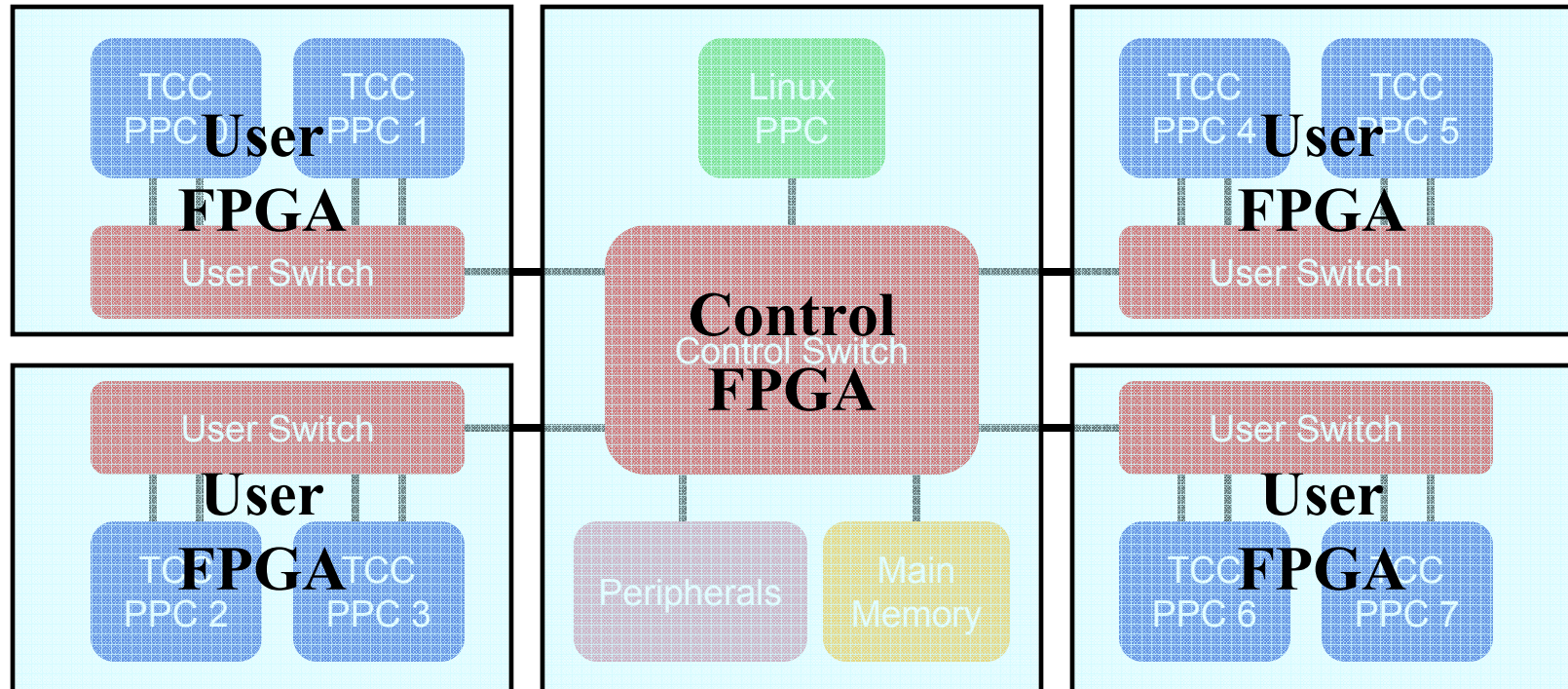
Serialize the system



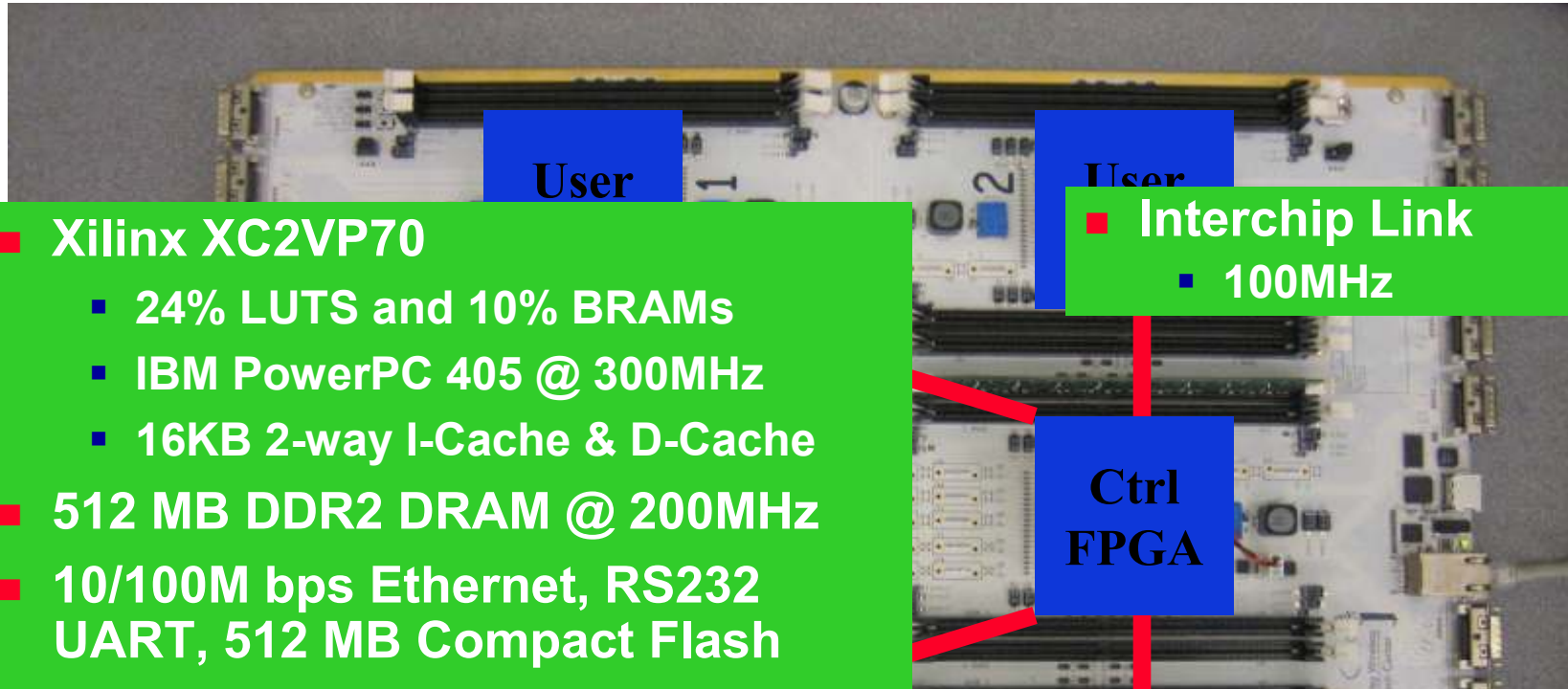
# ATLAS Architectural View



# ATLAS Hardware Mapping



# Mapping ATLAS on the BEE2



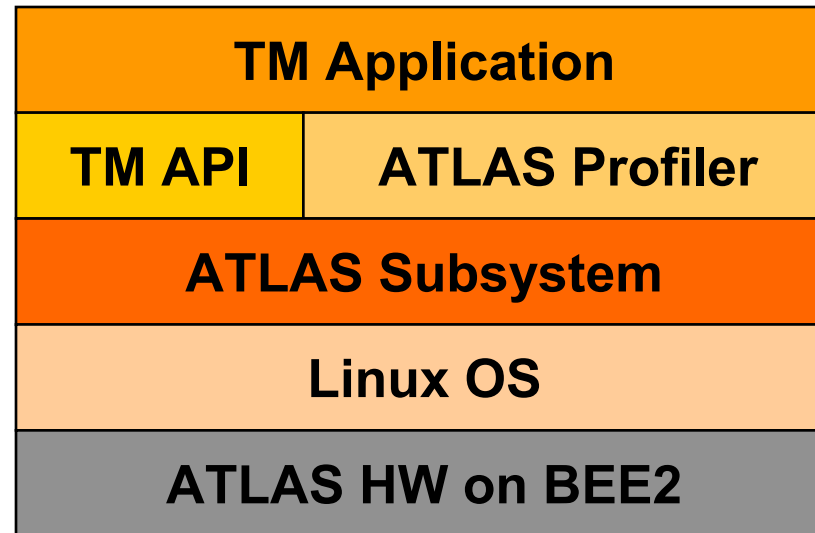
- **Xilinx XC2VP70**
  - 24% LUTS and 10% BRAMs
  - IBM PowerPC 405 @ 300MHz
  - 16KB 2-way I-Cache & D-Cache
- **512 MB DDR2 DRAM @ 200MHz**
- **10/100M bps Ethernet, RS232 UART, 512 MB Compact Flash**
- **MontaVista 3.1 Linux (kernel v2.4.30)**

- **Interchip Link**
  - 100MHz



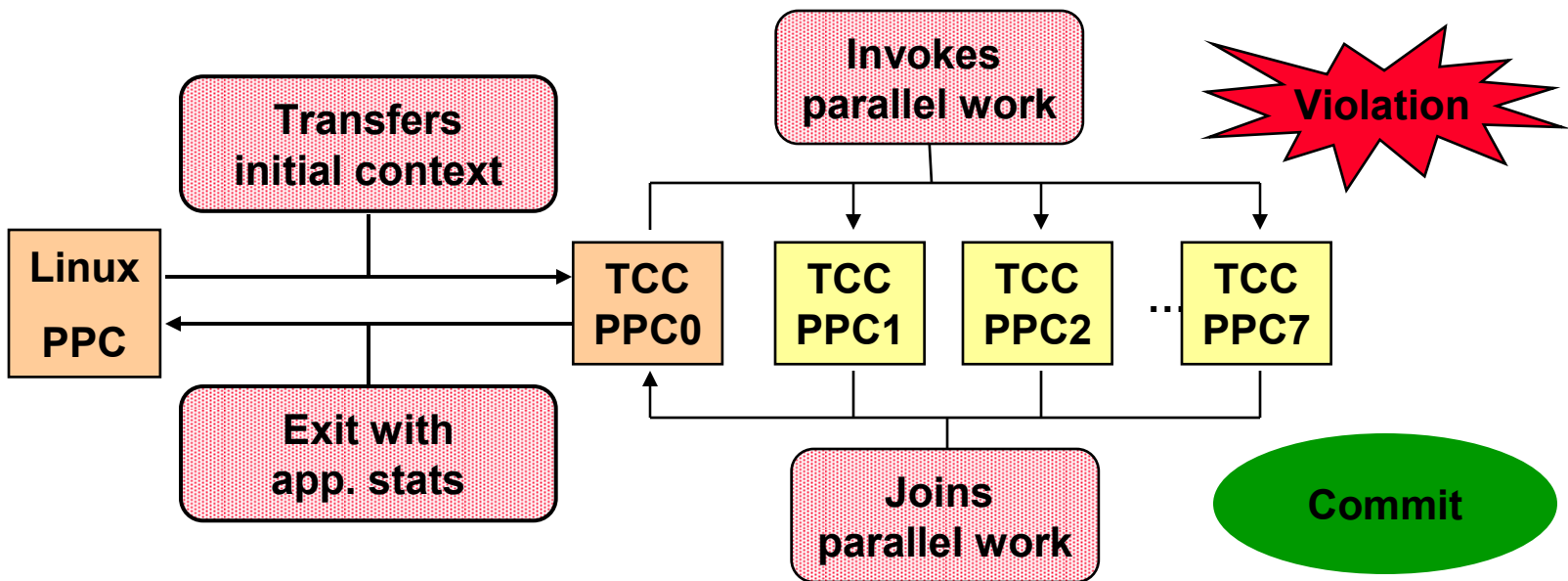
- **Xilinx XC2VP70**
  - 26% LUTS and 32% BRAMs
  - IBM PowerPC 405 @ 100MHz
  - 16KB 2-way I-Cache
  - Disabled internal D-Cache
  - 32KB 4-way D-Cache w/ TCC Support

# ATLAS Software Stack

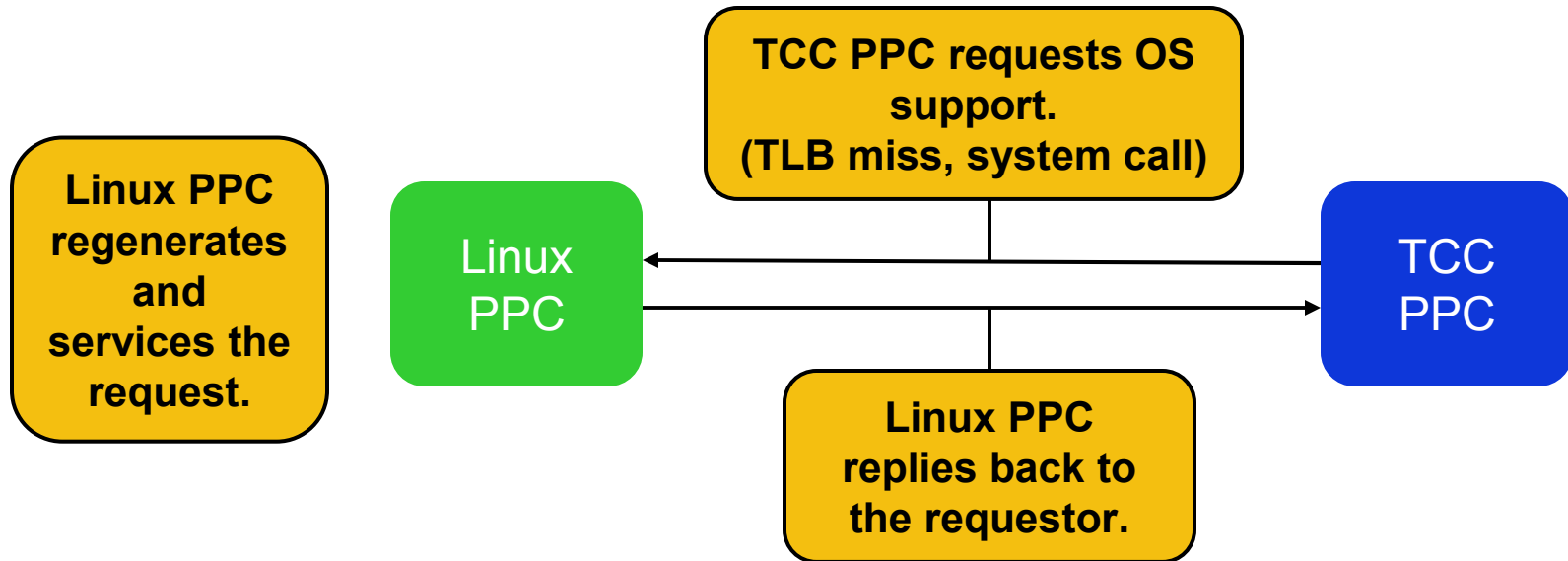


- **TM Application can be easily written with TM API.**
- **ATLAS Profiler provides a runtime profiling and guided performance tuning.**
- **ATLAS subsystem provides Linux OS support for the TM application.**

# ATLAS Subsystem Routine



# ATLAS Full System Support



- **Serialize for irrevocable requests.**
  - System Call
  - Page-out

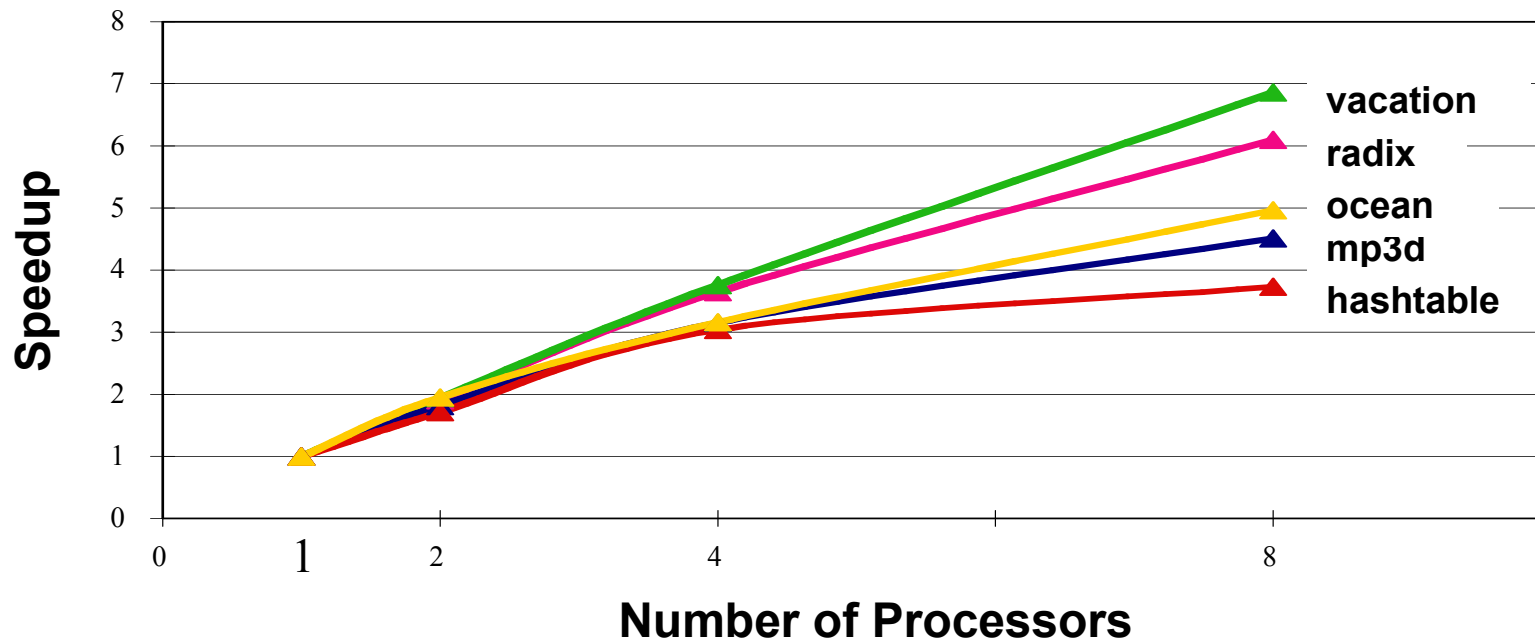


# ATLAS Runtime Profiler

- **TM Application's Performance Tuning point**
  - Reduce Violation due to the data conflict
  - Reduce Serialization due to the speculative buffer overflow
- **It is inspired by TAPE**
- **It records**
  - **Most Significant Violations**
    - Data address, Occurrence count, PC of the data access, Violator transaction, Wasted clock cycle.
  - **Most Significant Overflows**
    - Data address, Occurrence count, PC of the data access, Serialized clock cycle.

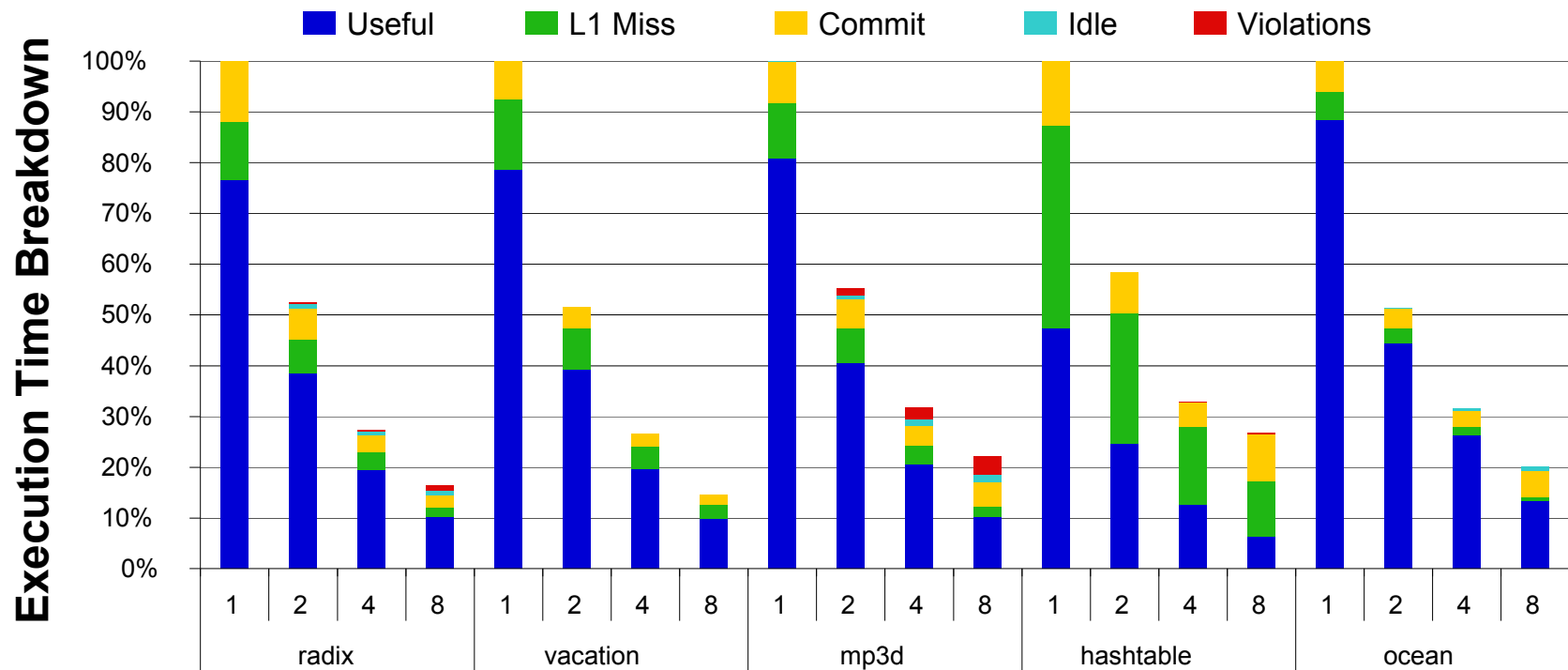
# Evaluation: Speedup

- **ATLAS shows that TM scales well with most of benchmarks.**



- **Some applications inherently does not perfectly scale.**
  - hashtable : Poor Locality
  - Mp3d: Data Dependency

# Evaluation: Visibility



**Applications  
with 1/2/4/8 processor configuration**

- **ATLAS provides easier way of application analysis.**
- **It matches well with our in-house software simulator.**
  - **100x Faster than the software simulator.**

# Conclusion

- **ATLAS is the first full-system prototype of a CMP with hardware transactional memory support.**
- **ATLAS shows that TM parallel program results good speedup performance.**
- **ATLAS provides fast software development platform with runtime performance profiling and guided tuning.**



# Thank You

**Your comments and questions are welcomed.**

**[tcc\\_fpga\\_xtreme@mailman.stanford.edu](mailto:tcc_fpga_xtreme@mailman.stanford.edu)**