

# NZTM: Nonblocking Zero-indirection Transactional Memory

Fuad Tabb<sup>1</sup>   Cong “James” Wang<sup>1</sup>  
James R. Goodman<sup>1</sup>   Mark Moir<sup>2</sup>

<sup>1</sup>Department of Computer Science  
University of Auckland

<sup>2</sup>Sun Microsystems Laboratories

The Second ACM SIGPLAN Workshop on Transactional  
Computing, 2007



# Outline

## Motivation

## NZSTM - Design and Implementation

- Data Structures

- Making NZSTM Nonblocking

- NZTM - The Hybrid Approach

## Our Results

- Test Environment

- Main Results

## Conclusion

- Ongoing and Future Work



# Why NZTM?

## Goal

Efficient nonblocking transactional memory.

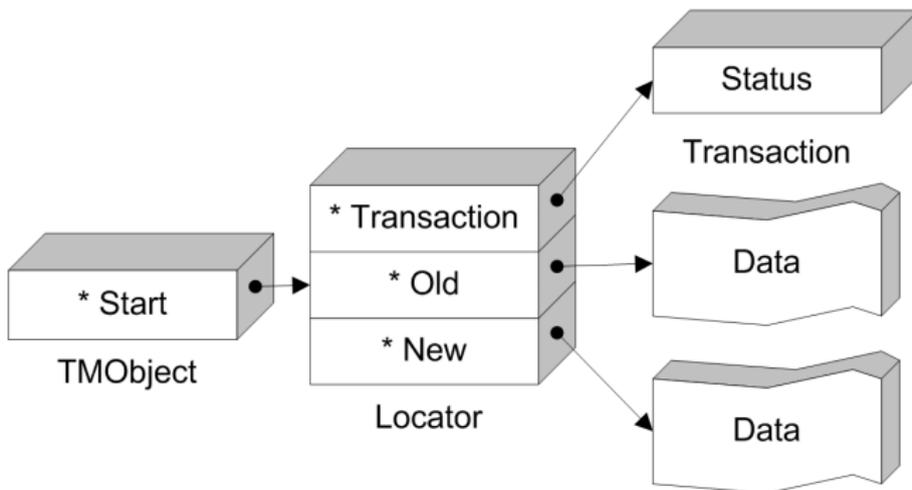
## Our Solution

- ▶ Object-based: suitable for object-oriented languages.
- ▶ Zero-Indirection: each level is a potential cache-miss.
- ▶ Hybrid: complete STM that can exploit best-effort HTM to improve performance.



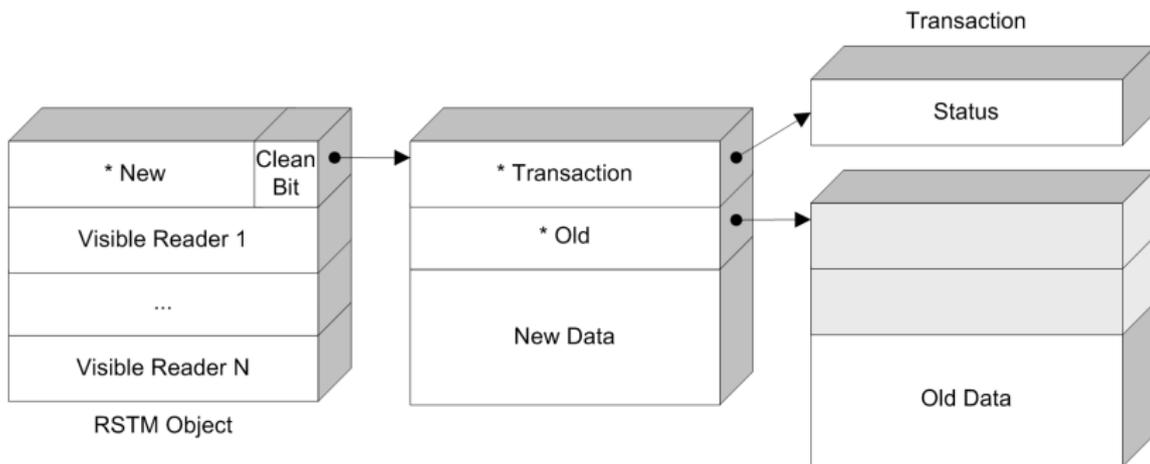
# DSTM

## TMObject



# RSTM

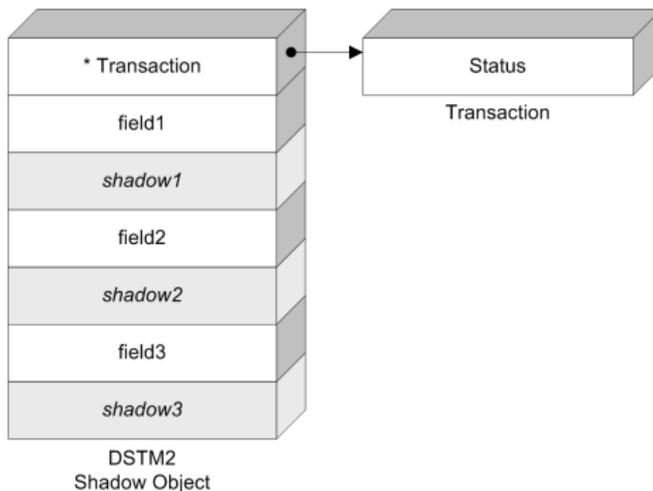
## RSTM Object





# DSTM2

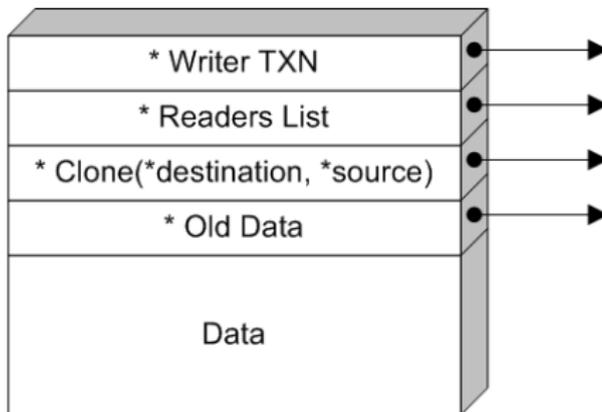
## DSTM2 Shadow Factory Object





# NZSTM

## NZObject

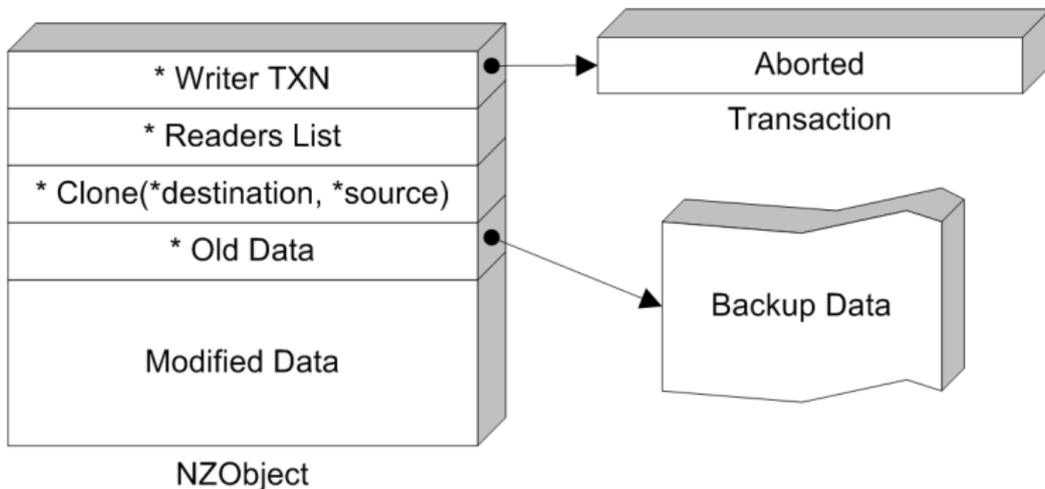


NZObject



# Transactions and Aborting

## Aborting an NZSTM Transaction





# Transactions and Aborting

## NZSTM Transaction

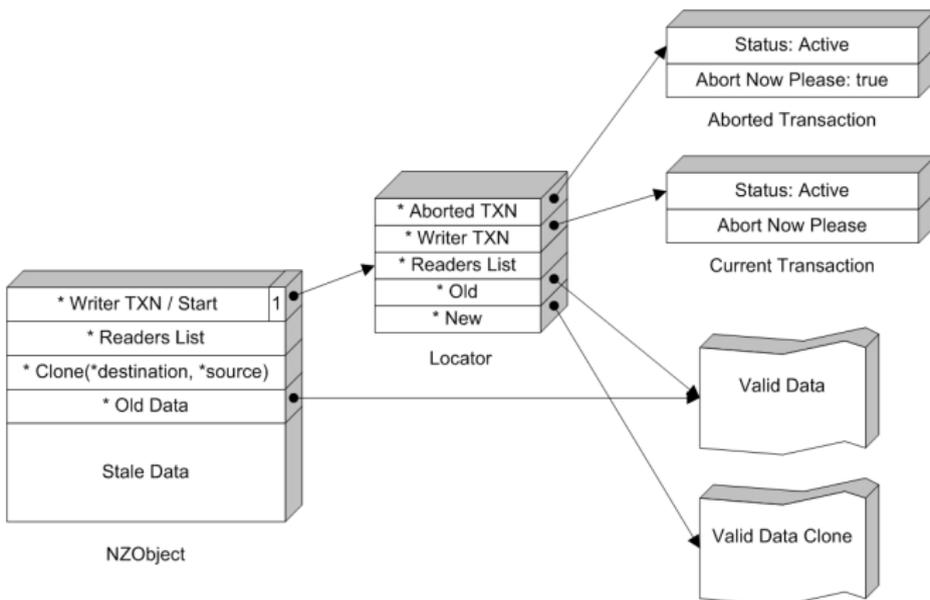


Transaction



# Making NZSTM Nonblocking

## Displacing Data



# NZTM - The Hybrid Approach

## Integrating Hardware and Software

- ▶ Compatible with best-effort hardware schemes.
- ▶ Attempts transactions in hardware first, then falls back onto software.
- ▶ We hope that hardware transactions would be the common case.



# Benchmarks and Experiments

## Benchmarks

- ▶ Linked List
- ▶ Red-black Tree
- ▶ Chained Hash Table
- ▶ Web Cache Simulation

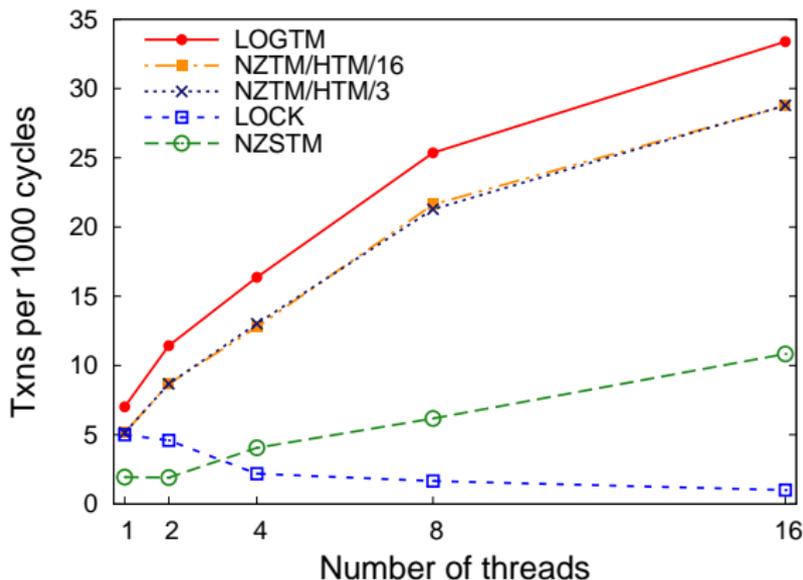
## Experiments

- ▶ Single Global Lock
- ▶ LogTM
- ▶ NZSTM
- ▶ NZTM



# Results

## Chained Hash Table





## Ongoing and Future Work

- ▶ More comprehensive evaluation.
- ▶ Investigate different design alternatives and optimizations.
- ▶ Currently implementing invisible reads.



# Making NZSTM Nonblocking - Alternative Method

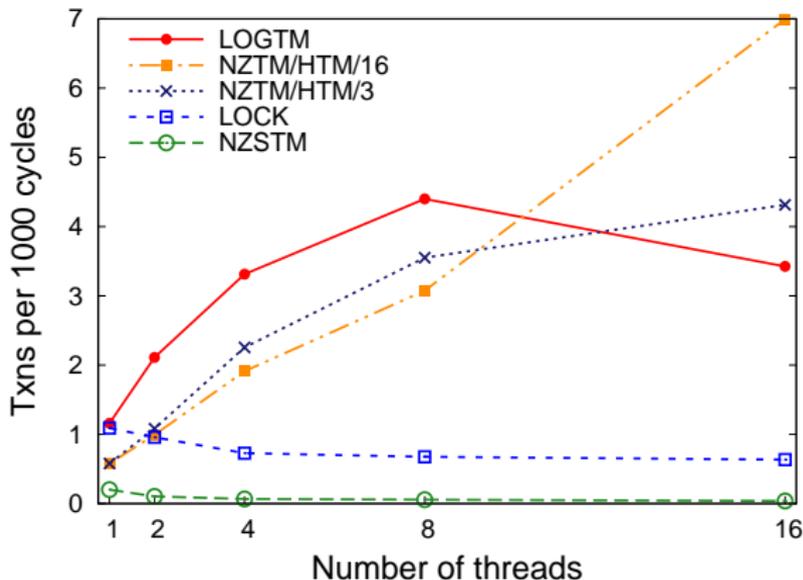
## Using Short Hardware Transactions per Store

- ▶ Begin Hardware Transaction
- ▶ Confirm transaction is *Active*
- ▶ Modify data in-place
- ▶ Commit Hardware Transaction



# Results

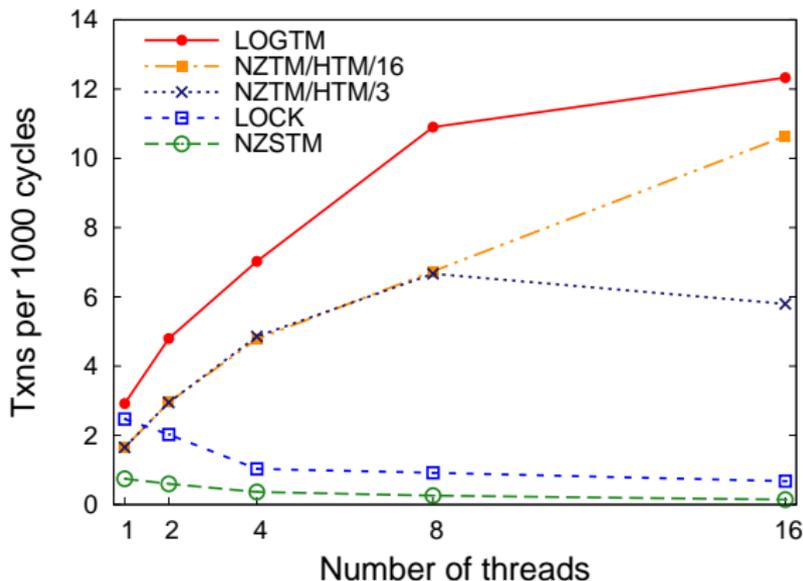
## Linked List





# Results

## Red-black Tree





# Results

## LFUCache

