MANCHESTER 1824

The University of Manchester

An Object-Aware Hardware Transactional Memory System

Behram Khan, Matthew Horsnell, Ian Rogers,

Mikel Lujan, Andrew Dinn and Ian Watson

http://www.cs.manchester.ac.uk/apt/projects/TM

Why?

- In 2006 more than 60% of new applications were using managed runtime environments (JVM, CLR, ...)
- In 2008 estimated to be more than 80%
- Current computer architectures only see a flat memory address space
- Communication infrastructure will be key in multi-core (many-core)
- We are interested in finding ways of reducing the demands on this infrastructure

In a Nutshell

- Similar to hardware support for paged virtual memory using virtually addressed caches and TLB
- Cache hierarchy addressing based on unique object identifiers
- Benefits:
  - Support of cache overflows of uncommitted data
  - Novel commit and conflict detection mechanisms
- Object granularity / Lazy versioning / Lazy conflict detection

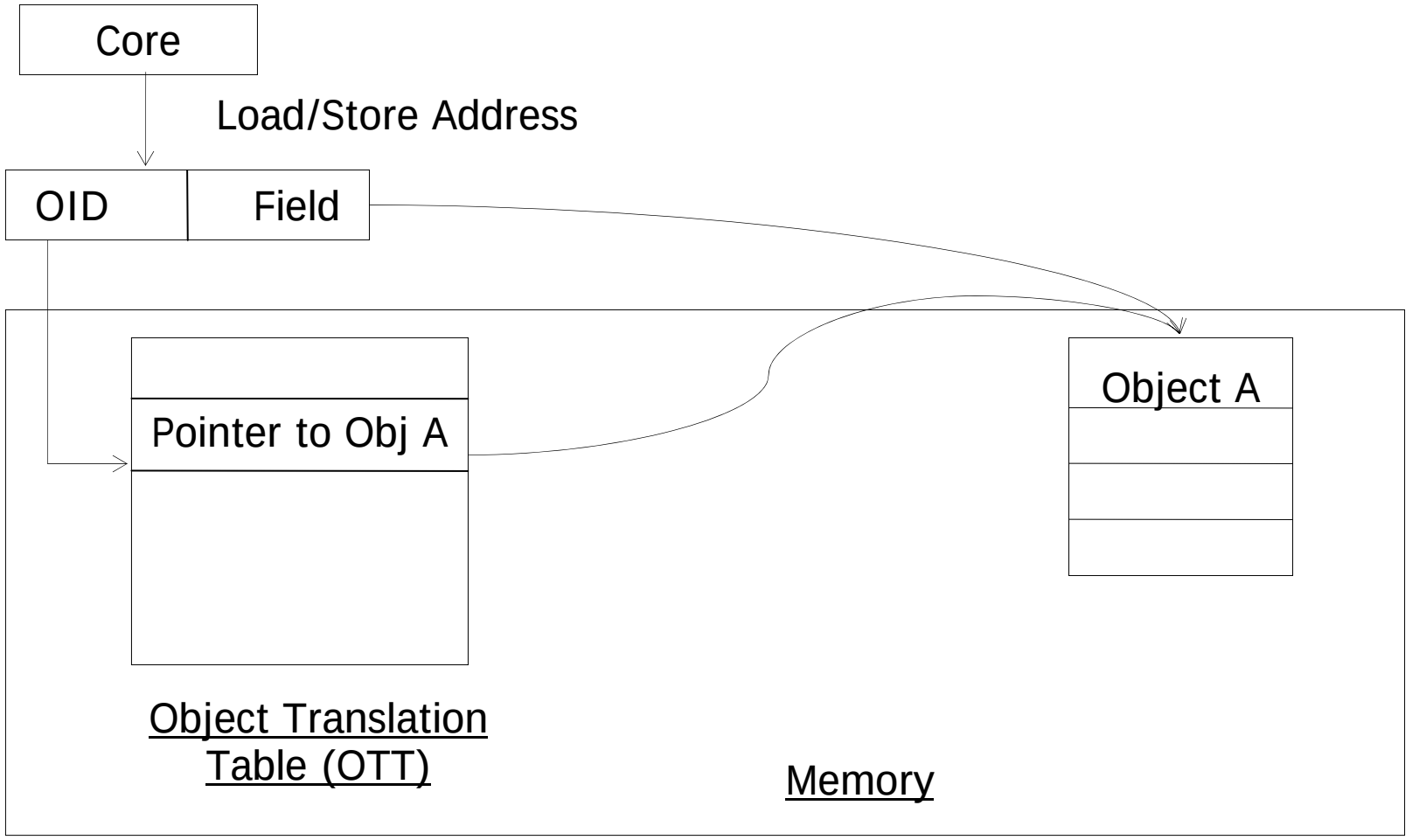
Hardware Transactional Memory

- Buffer changes in hardware
- Exploit buffering already present in cache (TCC)
- Or create undo log and assume conflict is unlikely (LogTM)
- Resort to STM when buffers/logs overflow
- Detect conflicts at the cache line level

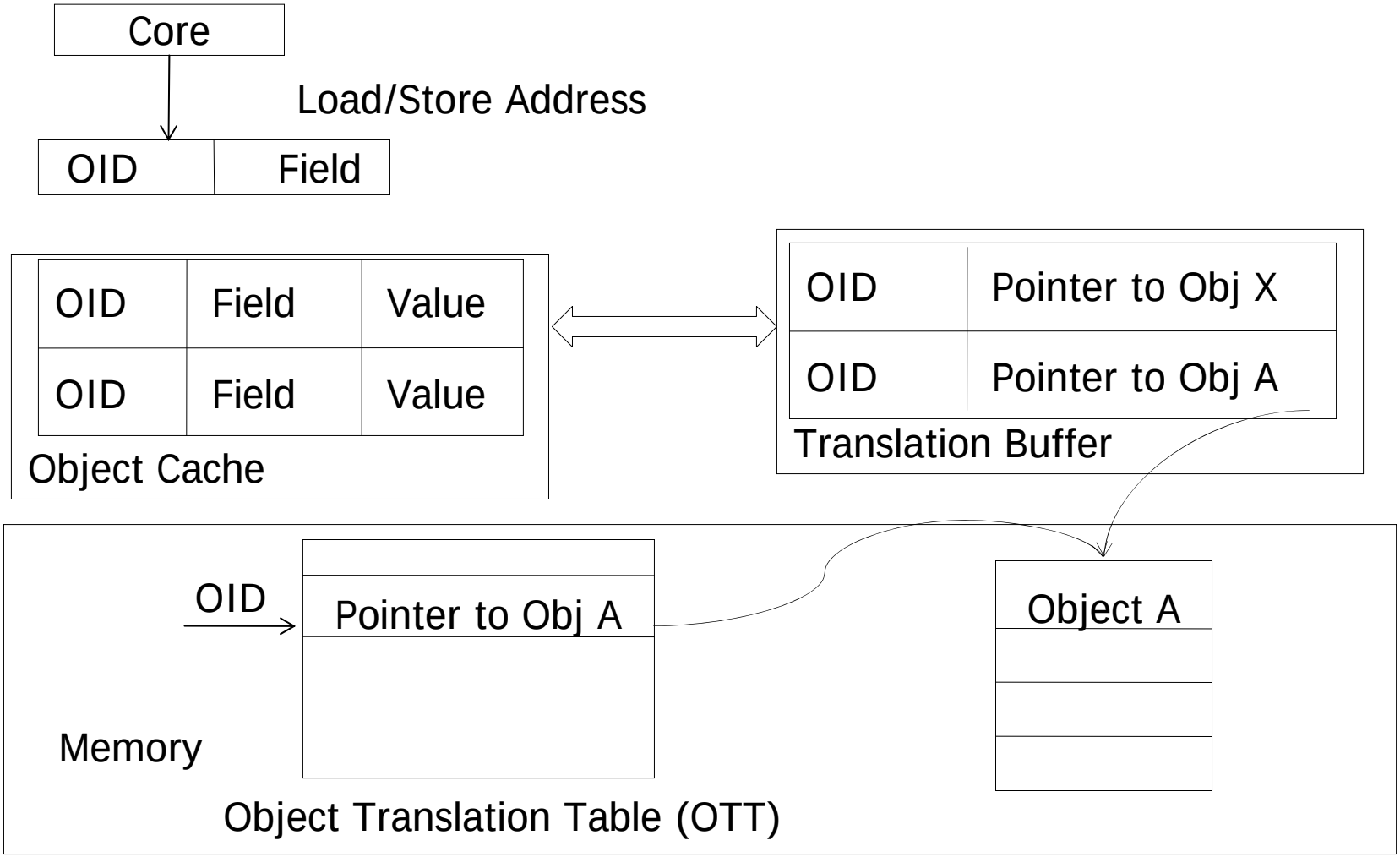
Problems

- TM proposals artificially work at granularities of existing memory systems
  - byte, short, int, long
  - cache line
  - page
- Often assume that conflicts are infrequent
  - rollback is expensive
- There is a lack of harmony between the world of the programmer (objects) and the hardware

Object-Aware Memory Architecture



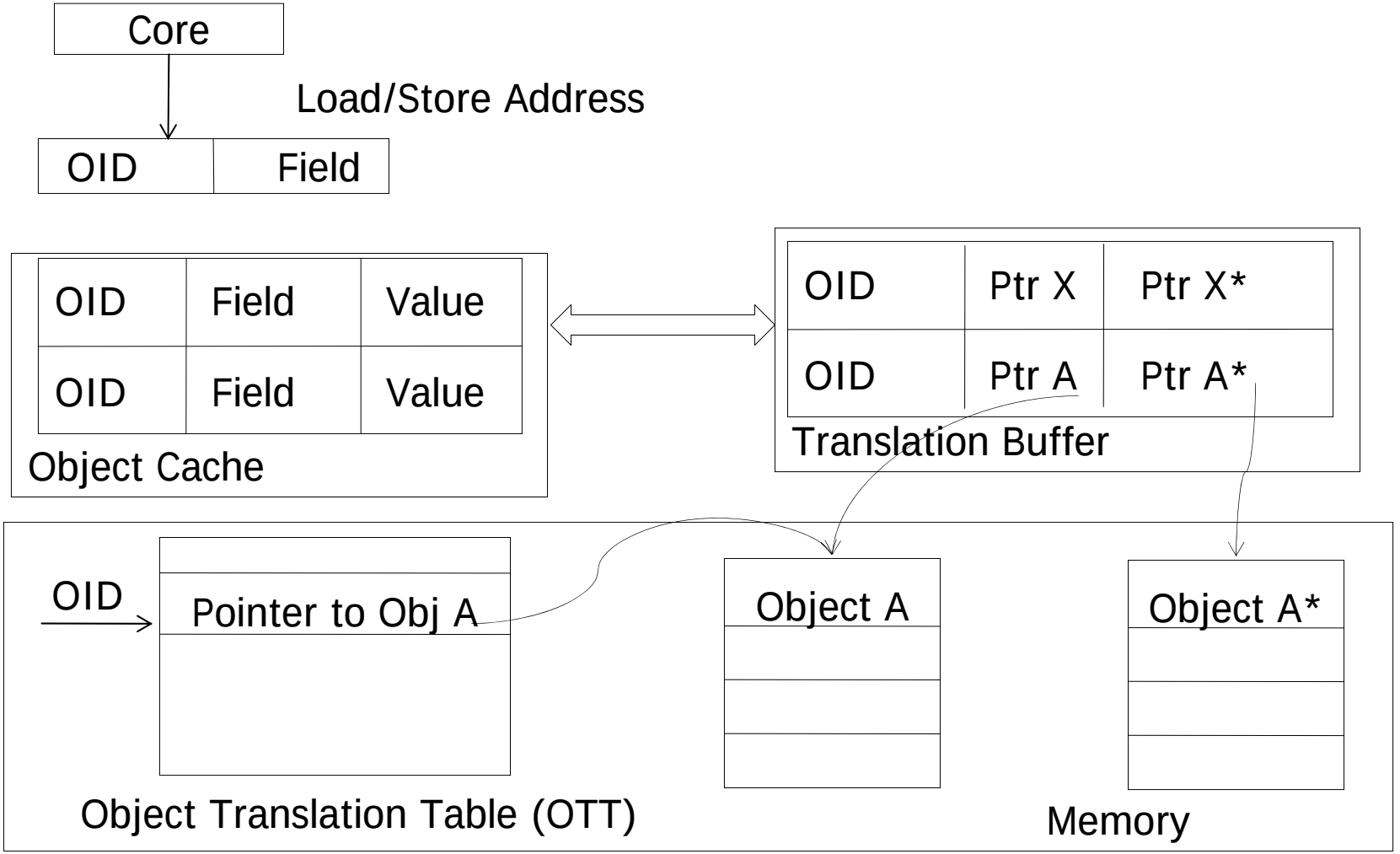
Object-Aware Memory Architecture



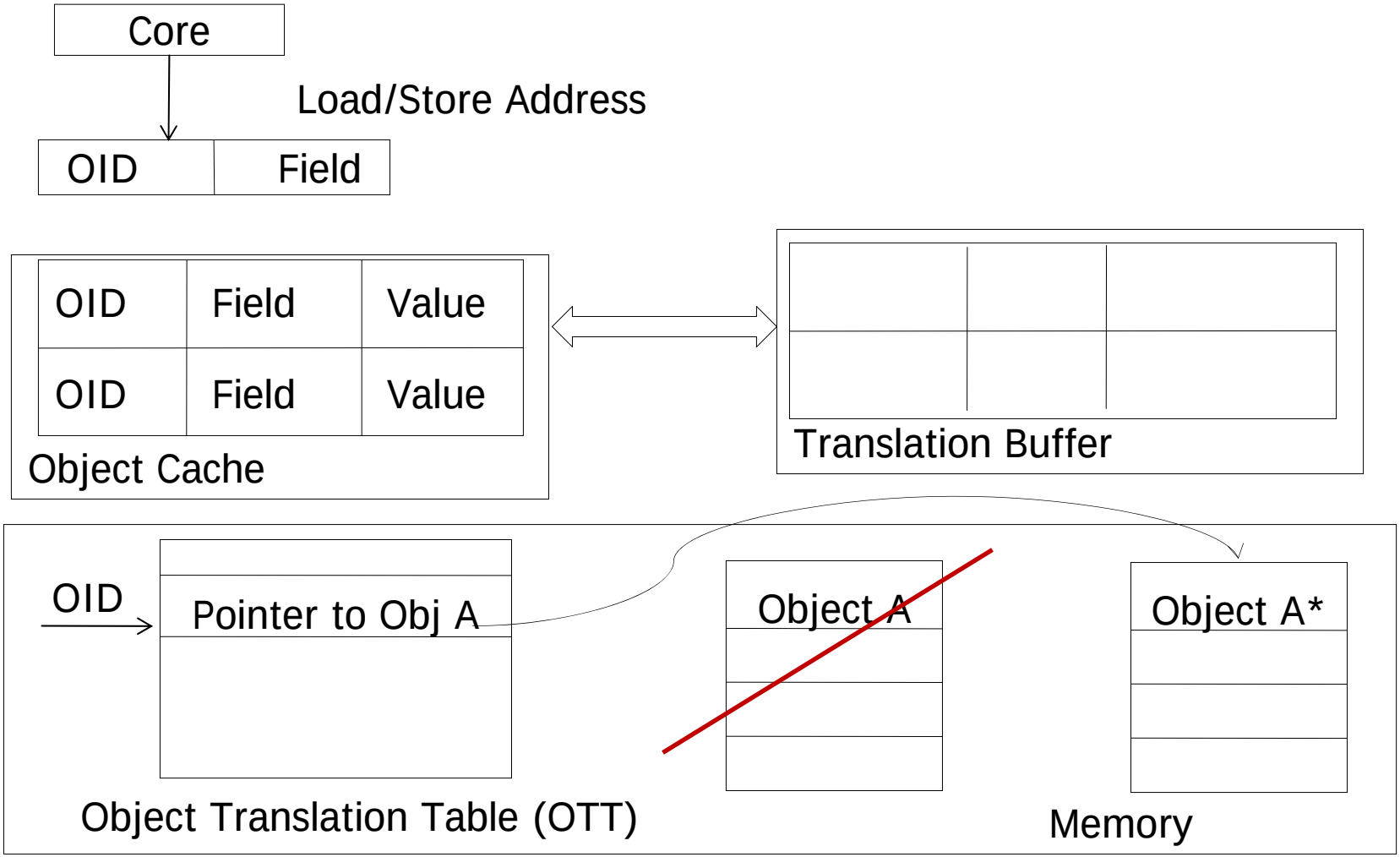
What's been achieved

- Memory now addressed by objects
- Virtually addressed cache
- Large objects map to consecutive OIDs
- Using objects allows aspects of protocol to be concise whilst avoiding false sharing
- Indirection has been shown to aid garbage collection
  - Moving an object requires a single pointer update

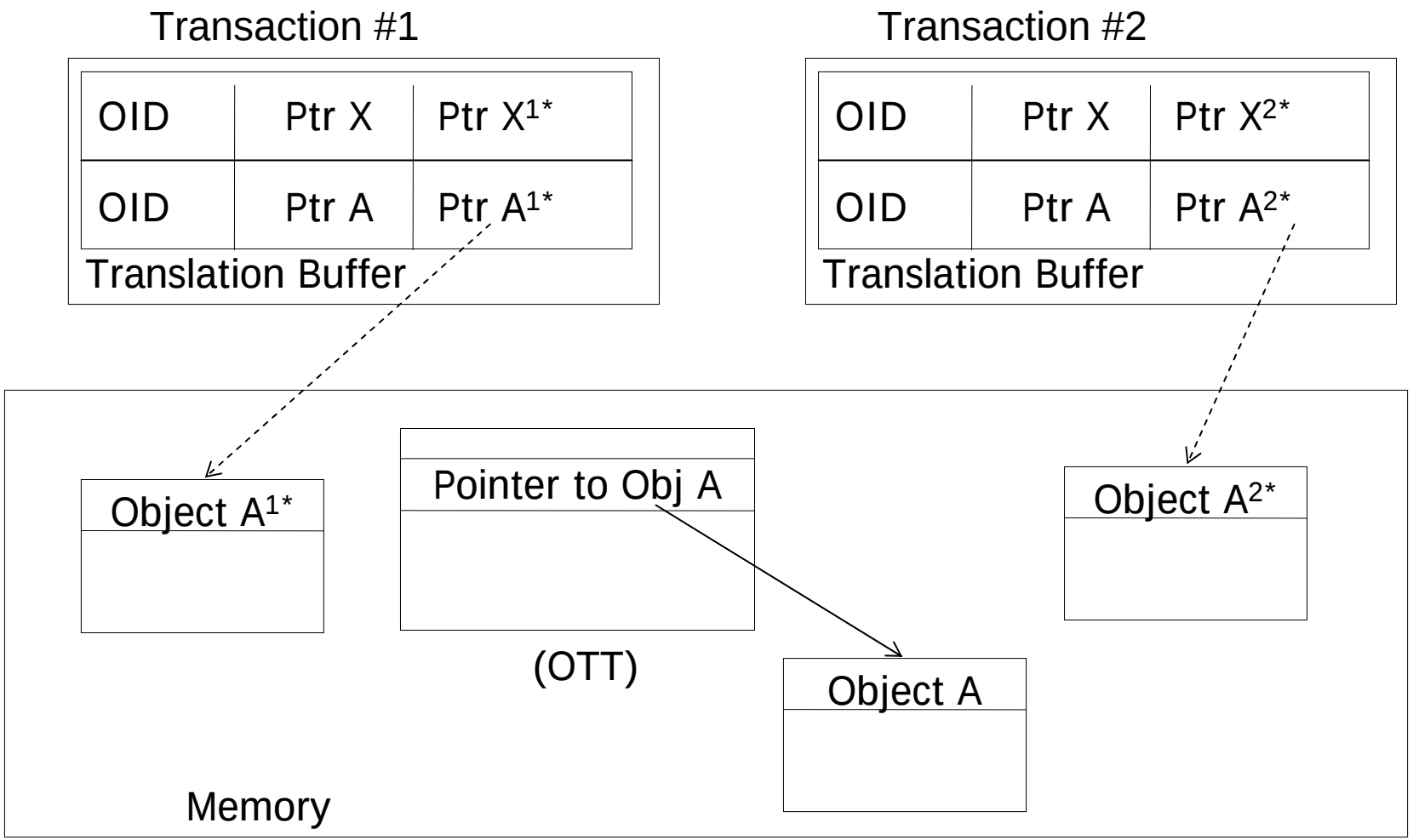
Object-Aware Transactional Memory - during transaction



Object-Aware Transactional Memory - commit



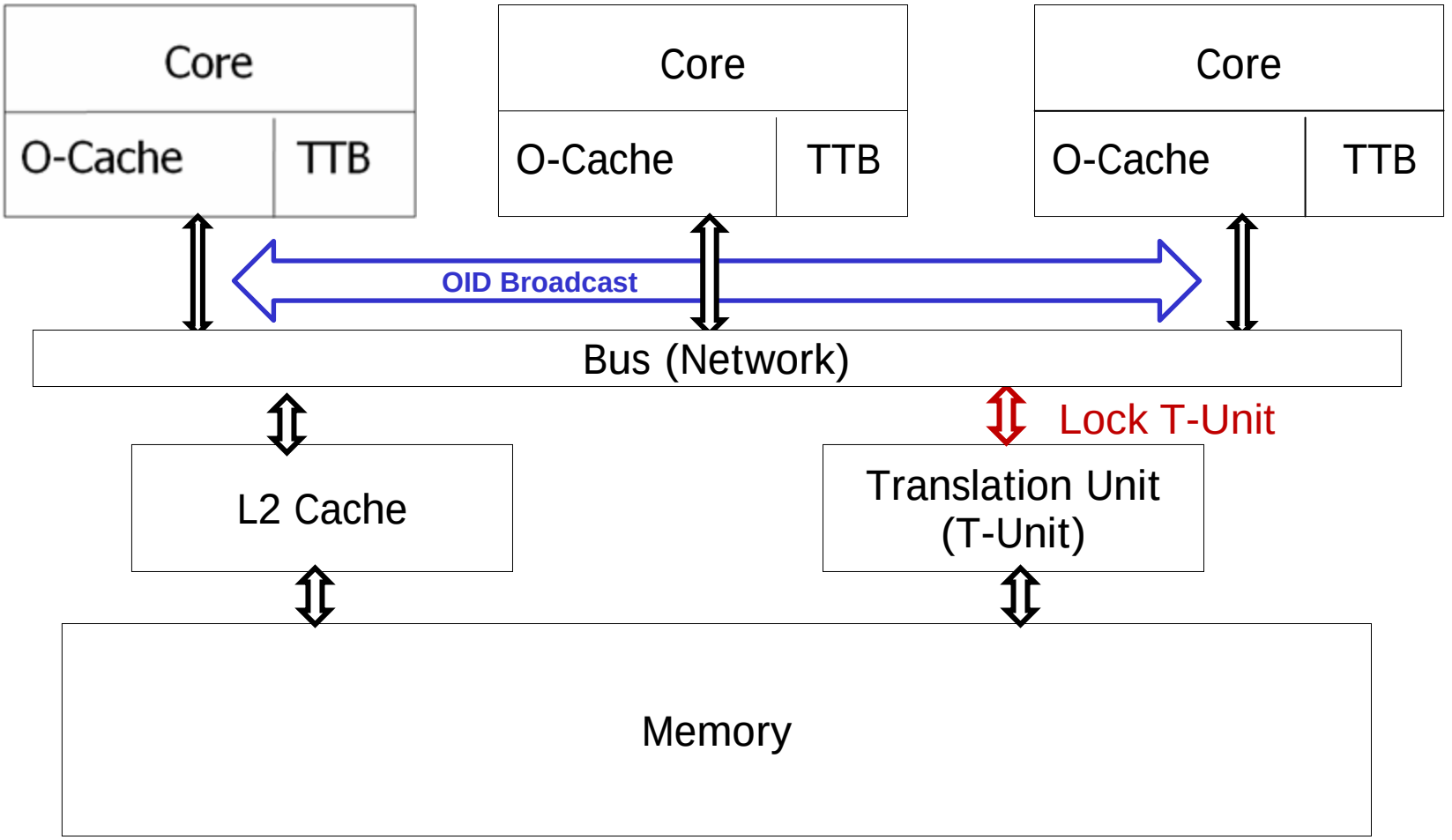
Object-Aware Transactional Memory - conflict



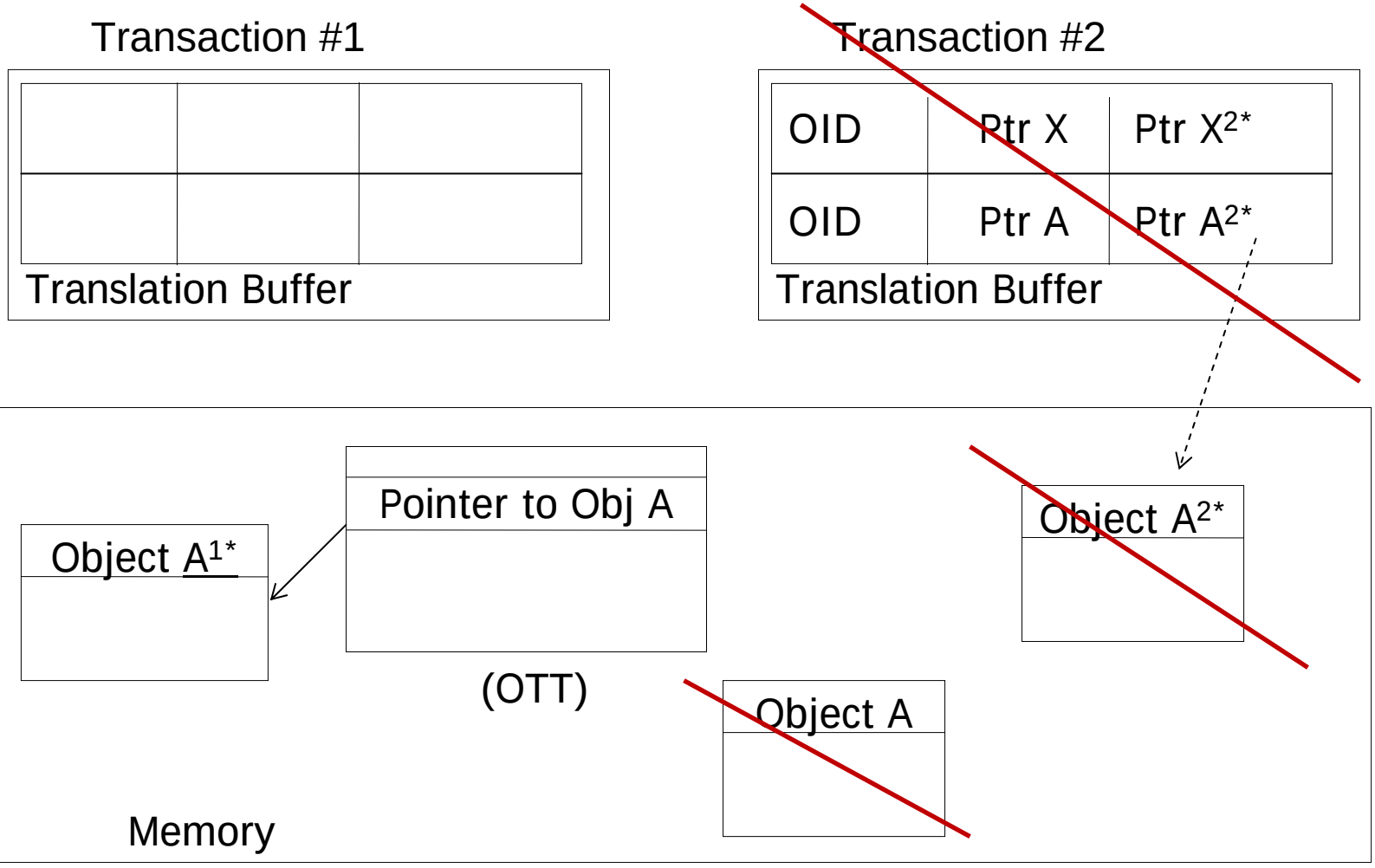
MANCHESTER 1824

The University of Manchester

Object-Aware Transactional Memory



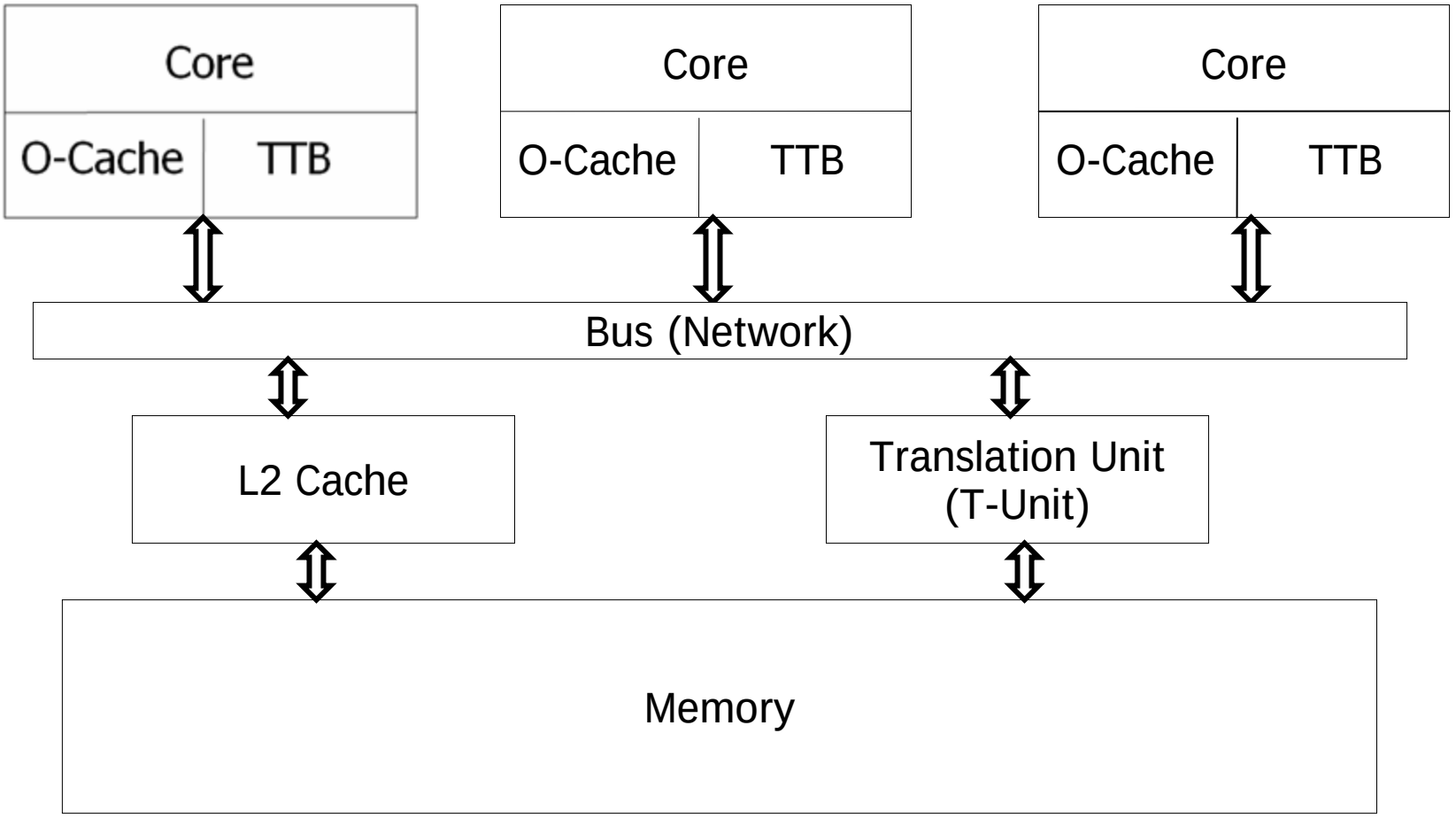
Object-Aware Transactional Memory - conflict detected by broadcast of OID



MANCHESTER 1824

The University of Manchester

Simulated Object-Aware Transactional Memory



MANCHESTER 1824

The University of Manchester

Evaluation

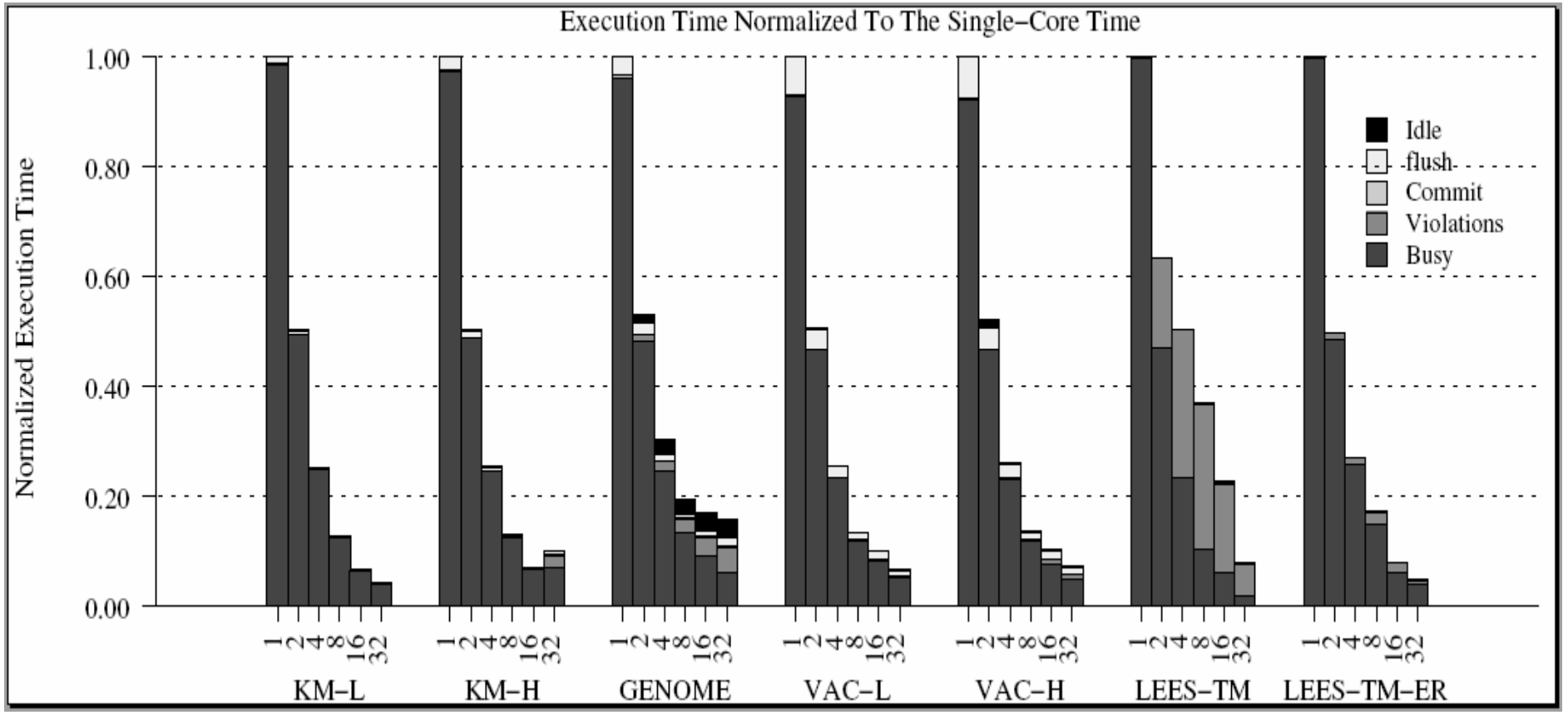
Feature	Description
Object size	128B.
L1 object cache	32KB, private, 4-way assoc, 32B line, 1-cycle access.
TTB	24KB, private, 4-way assoc, 12B lines, 1-cycle access.
Network	256-bit bus, split-transactions, pipelined, no coherence.
L2 cache	4MB, shared 32-way assoc, 32B lines, 16-cycles access.
TU	4MB, shared, 32-way assoc, 12B lines, 16-cycles access.
Memory	100-cycles off-chip access.

Simulation parameters

MANCHESTER 1824

The University of Manchester

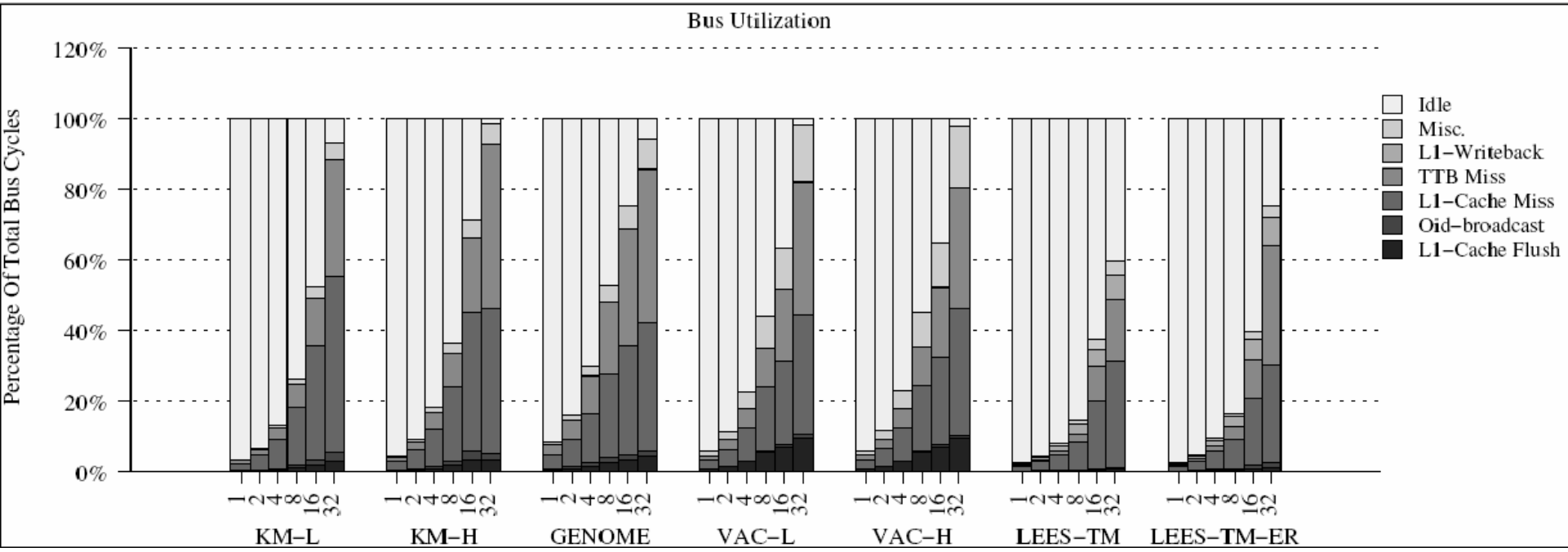
Normalized Execution Time



MANCHESTER 1824

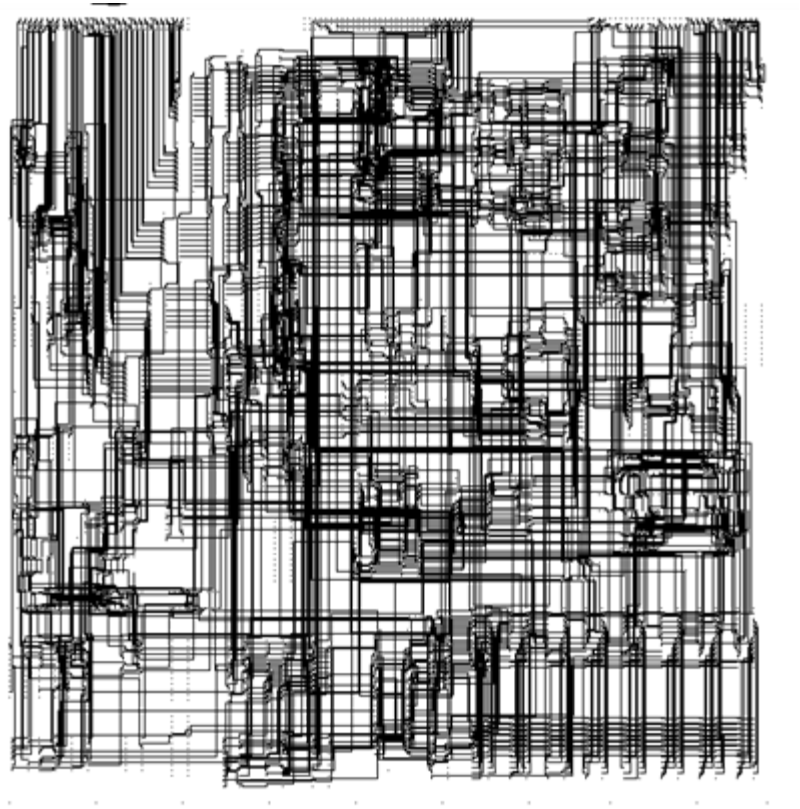
The University of Manchester

Breakdown of Bus Traffic



What is Lee-TM?

- Realistic TM benchmark
- Benchmark based on Lee's routing algorithm
  - Used in circuit routing
  - Each route is pair of points
  - Connect pair of points
    - Shortest distance possible
    - Can't cross other routes
  - Connection on 3D array
    - Represents circuit board



MANCHESTER 1824

The University of Manchester

Evaluation

Application	Readset/Tx		Writeset/Tx		Inst/TX		Overflow	
	Mean	CoV	Mean	CoV	Mean	CoV	Txs	Lines
LeeTM	117.5	2.9	78.8	3.1	373810.6	4.1	287	407255
LeeTM-ER	15.8	1.4	78.8	3.1	373813.9	4.1	291	407683

Lee profile

Application	L1-Cache Flush	Oid-broadcast	L1-Cache Miss	TTB Miss	L1-Wrback	Misc	Idle
LeeTM	.8%	.5%	30.1%	17.4%	7%	4%	40%
LeeTM-ER	1.2%	1.3%	27.5%	33.9%	8%	3%	24%

Bus Utilization

In a Nutshell

- Similar to hardware support for paged virtual memory using virtually addressed caches and TLB
- Cache hierarchy addressing based on unique object identifiers
- Benefits:
  - Support of cache overflows of uncommitted data
  - Novel commit and conflict detection mechanisms
- Object granularity / Lazy versioning / Lazy conflict detection

MANCHESTER 1824

The University of Manchester

Summary

- Object-Aware Hardware Transactional Memory
  - Elegantly allows cache overflows of uncommitted data
  - Deal with frequent aborts (rollback overhead)
  - Reduce the demands on network-on-chip
    - OID broadcast accounts for lees than 1.5% of the total bus traffic using LeeTM and LeeTM_ER benchmarks.
  - Concise handling of programmatic entities
  - Benefits to runtime and garbage collection

MANCHESTER 1824

The University of Manchester

Backup Slides

Extensions

- Bitmaps within object
  - prevent false sharing within an object
- Self validation
  - Suspending (even migrating) transactions can be checked to see if cached object reference is the same as in the OTT
- Overflow
  - If caches and TTB overflow then parts can be written to memory - a bloom filter encoding OIDs can be used to conservatively approximate the TTB entries that have been spilled to memory

Speedup Results