Cycle-Accurate Memory Simulation: a Case Study in Validation

Andrew Over, Peter Strazdins*
and Bill Clarke

CC-NUMA Project,
Department of Computer Science,
The Australian National University

MASCOTS 2005, 27 September 2005

http://cs.anu.edu.au/~Peter.Strazdins/seminars#memsim

THE AUSTRALIAN NATIONAL UNIVERSITY

1 Overview

- performance evaluation of memory-intensive applications on SMPs
- the Sun Microsystems V1280 FirePlane memory system & its protocols
- execution-driven simulation: overview and issues
- traditional SMP simulator design
- design: modelling of the memory system
  - use of PDES, processor model modifications, simulator re-structuring
  - modified simulator run loop and pipelined transactions
- validation
  - micro-benchmarks
  - OpenMP NAS Parallel Benchmarks & use of hardware event counters
- performance & accuracy impact of detailed memory system modelling
- conclusions and future work

2 Performance Evaluation of Memory-Intensive Applications

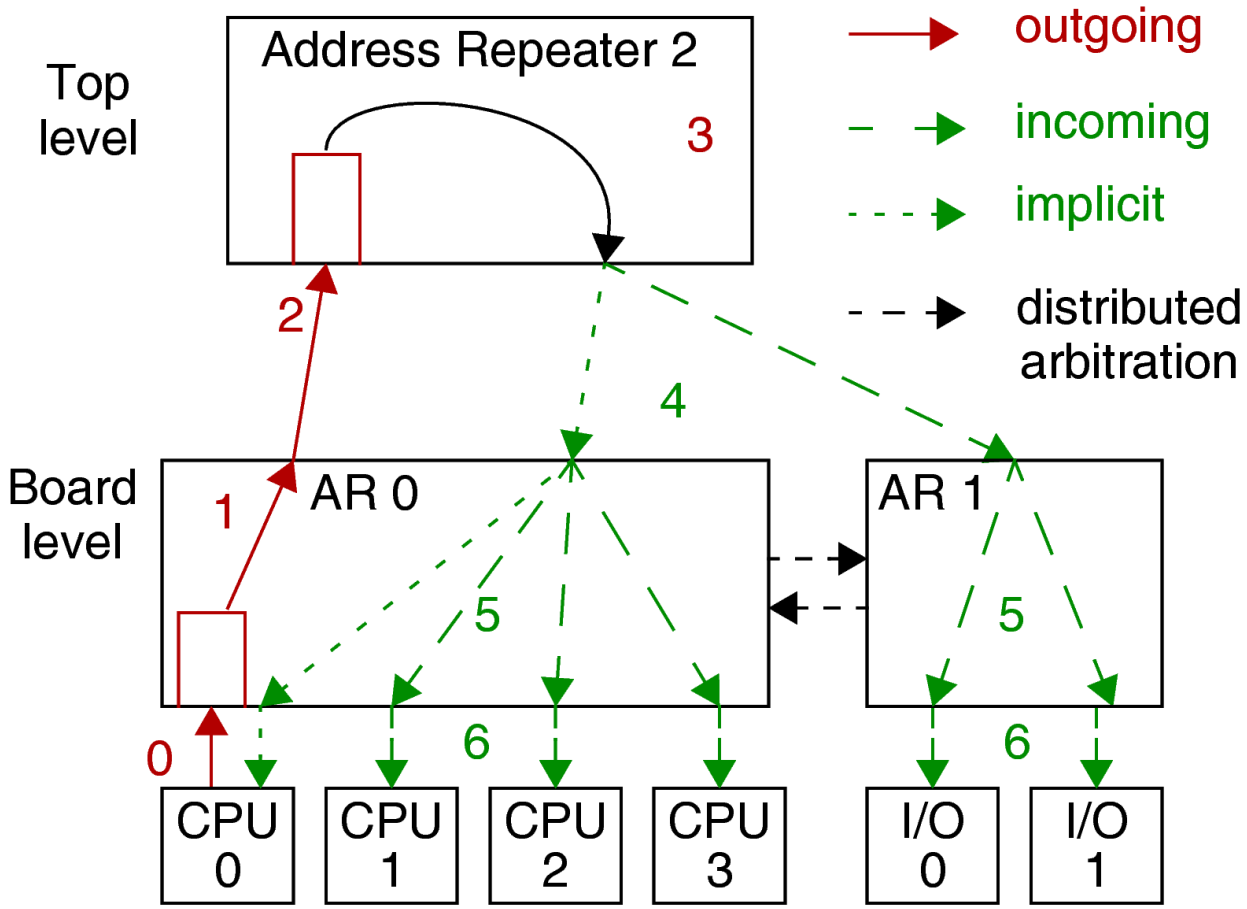
- large-scale symmetric multiprocessors with cc-NUMA memory systems are important HPC platforms
- as well as commercial applications, many scientific applications are limited by memory performance
- e.g. in computational chemistry, linear scaling algorithms, as used in applications such as Gaussian:
  - most activity is at user-level; memory intensive
  - irregular memory accesses, limited temporal locality
  - parallelize with special emphasis on data placement
  - thread affinity issues also important
  - realistic analysis requires large workloads!

  the CC-NUMA Project is concerned with such analysis
- detailed and accurate simulation of these memory systems can provide valuable insights into performance bottlenecks on current and (similar) future systems

3 Target Architecture: the UltraSPARC IIICu V1280

- Sun Microsystems donated a 12 CPU (900 MHz) UltraSPARC V1280 to the ANU
  - 32KB I-Cache, 64KB D-Cache, 8MB E-cache
  - relies on hardware/software prefetch for performance (≤ 8 outstanding transactions)
  - total store ordering
  - 2KB W-Cache, 2KB P-Cache, 8 entry store buffer
- Sun FirePlane interconnect at 150 MHz
  - single snooping coherence domain
  - integrated support for FirePlane coherency protocol
  - 2 CPU pairs per board, address repeaters on boards connected to top-level address repeater
  - data bus uses a multilevel bi-directional crossbar network
- physical access to this system allows detailed benchmarking and provides a timing reference

4 Target Architecture – FirePlane Protocol



FirePlane address bus timing (from Alan Charlesworth, The Sun Fireplane System Interconnect, *ACM/IEEE Conference on Supercomputing*, Nov 2001.)

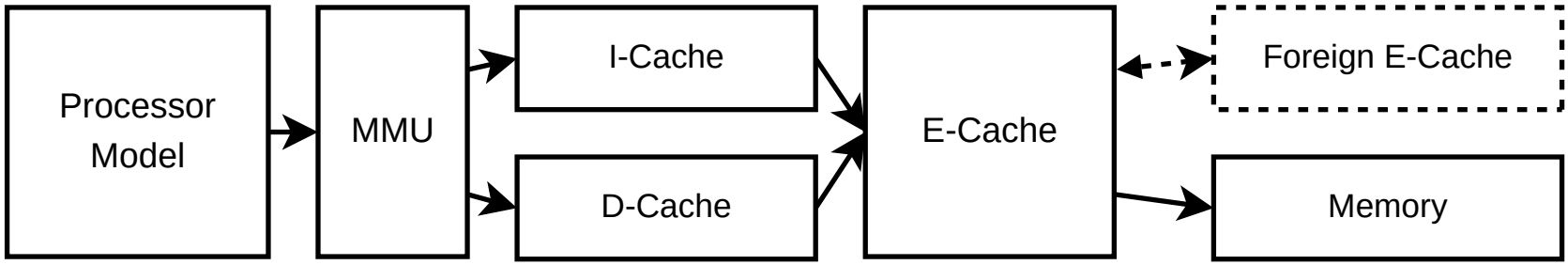
5 Target Architecture – FirePlane Protocol (cont)

1. CPU 0 begins transaction (cycle 0)
2. system address repeater broadcasts address (cycle 3)
3. all CPUs snoop transaction address (cycle 7)
   CPUs respond (cycle 11)
   CPUs see result (cycle 15)
4. owner initiates memory request (cycle 7)
5. data transfer begins (cycle 23)
   - completion varies depending on distance
   - 5 cycles for same CPU
   - 9 cycles for same board
   - 14 cycles for different board

note: here 'CPU' means 'the E-cache associated with the respective CPU'

6 Existing Simulation Techniques – Execution Based

- mimics a real processor in software
  - captures both functional simulation *and* timing simulation
- Fetch/Decode/Execute is a simple & reasonably fast Processor Model
- a simple hierarchical memory system is typically used:
  - fixed latencies (models blocking caches)
  - can model SMPs with round-robin time-slicing
  - instantaneous access/update of other processor's cache lines



- challenge: adapt to this accurately model a modern NUMA system

7 Challenges in Modelling cc-NUMA Memory Systems

- timing critical in behaviour of SMP systems (hence execution-driven approach)
  - round-robin time-slicing leads to unrealistic memory event interleavings
- processor speed has improved much faster than memory latency
- memory latency hiding is crucial to performance of modern systems
  - multiple overlapping transactions are possible
  - over store queue, prefetch cache and FirePlane
- cache coherency mechanisms also introduce significant overhead
  - exclusive access to data is required before modification
  - writes must be carefully ordered
  - stale data must either be updated or invalidated
- interconnect contention and NUMA-related effects are important

8 Parallel Discrete Event Simulation

- lookahead (conservative PDES)
  - determine minimum latency between elements of system
  - need only synchronize between elements in accordance with this latency ("lookahead")
- minimum latency between effect and impact on foreign CPU in the FirePlane:
  - 7 cycles between snoop request and snoop response
  - 9+ cycles for inter-CPU data transfer

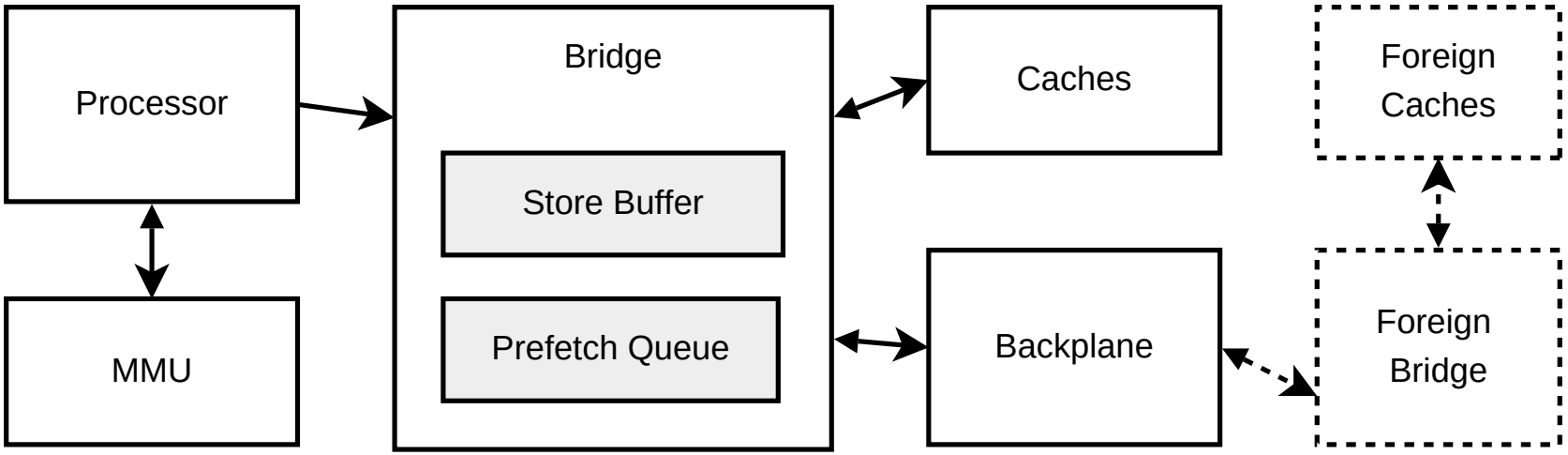
  $\Rightarrow$ can use a lookahead timeslice of up to 7 (bus) cycles

  (= 7*6 CPU cycles)
  - currently we use 4
- if processor is aware of all events occurring in past timeslice, need not communicate with other processors until simulation of timeslice completes
- potential for parallelisation (speedup on SMP host)

9 Design: Simulated Processor Model Modifications

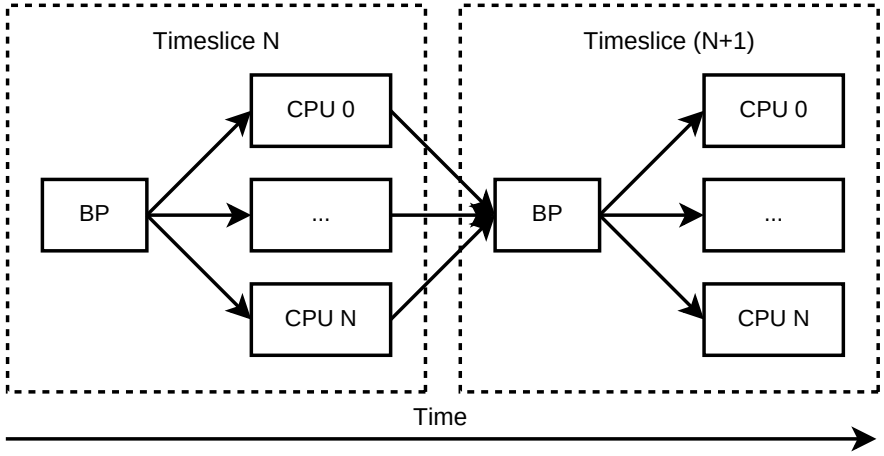
- integrate an accurate memory model in an existing (SimOS/Mipsy-like) UltreSPARC simulator
  - Fetch/Decode/Execute Processor Model
- as the latency of memory transactions is not known, introduce a fake "stall exception"
  - causes retry of current (load/store) instruction after a time period has elapsed
  - used to "poll" memory system for transaction completion
- add an event queue to the processor model
  - process any snoop events (request and response)
  - allows asynchronous polling of transactions outside instruction stream (prefetch, store buffer)
- an efficient cycle-accurate CPU timing module is added

10 Design: Simulator Restructuring



- introduce a bridge to isolate processor model from memory model
  - this handles all memory transactions
  - the MMU is used only for address translation
  - the backplane is used to model the interconnection network
  - all communication between processors is isolated through backplane
- structure allows easy substitution of different (simpler) memory models

11 Design: Run Loop and Pipelined Transactions



- new simulator 'run loop'
  - simulate backplane for (previous and) current timeslice
    - add coming events to the processor's event queue
  - run processors (serially or in parallel) for timeslice
    - must check their queue upon each simulated cycle
- a memory request data structure tracks progress of a transaction
- prefetch queue required careful implementation (invalidate on snoop events, merge overlapping loads)

12 Validation

- verifying simulator accuracy is critical for useful performance analysis
- validation is an ongoing issue in field of simulation
  - approximations or time constraints complicate comparison
  - do not necessarily have a physical model of system
  - validation only against 'cycle-accurate' simulators is common
- microbenchmarks:
  - allows verification of single memory transactions & easy to write
- application-level: by the OpenMP version of the NAS Parallel Benchmarks
  - use of hardware event counters (via UltraSPARC CPC library)
  - √ permits a deeper-level of validation than mere execution time
  - √ also provides breakdown of stall cycles (e.g. CPU, D/E-cache miss, store buffer)
  - × hardware counters are not 100% accurate; also ambiguously/incompletely specified (e.g. stall cycle attribution)

13 Validation: Microbenchmarks

- e.g. cache-to-cache transfer microbenchmark:

Processor A

```
1:  st      %g0, [A]
    call    _barrier


    call    _barrier
    ba      1b
```

Processor B

```
1:  call    _cache_flush
    call    _barrier
    rd      %tick, %l0
    ld      [A], %g0
    rd      %tick, %l1
    sub     %l1, %l0, %l0
    call    _barrier
    ba      1b
```

- also D/E Cache load/store hit/miss (various cache states/CPU pairs), atomic instr'n latency, store bandwidth, memory access (various regions), RAW, etc
- preferable to (possibly erroneous, out-of-date) data sheets
- provided valuable data, with several surprising & undocumented effects

14 Validation: NAS Benchmarks (S-class)

- p threads; number of cycles target: simulator (% of Total) (new)

Metric (p)	BT	FT	IS	LU	LU-hp	MG	SP
DC_miss	0.88 5%	0.44 12%	0.97 18%	0.44 10%	1.01 13%	1.13 31%	0.91 22%
SB_stall	1.20 27%	0.93 41%	1.15 54%	0.80 4%	0.84 14%	1.17 2%	0.72 14%
Total (1)	1.06	0.85	1.11	1.03	1.00	0.93	0.97
Total (2)	1.05	0.78	1.10	1.00	1.00	0.89	0.93
Total (4)	1.03	0.72	1.17	1.01	1.28	1.02	0.85
EC_miss	0.16 3%	0.13 4%	0.33 5%	0.12 8%	0.27 19%	0.28 11%	0.20 9%
SB_stall	1.22 27%	0.67 36%	1.22 47%	0.64 9%	0.69 19%	0.45 11%	0.64 19%

- simulator accuracy reasonable ($p = 1$), but less accurate as p increases
  - E-cache miss cycles consistently underestimated (possibly target is including cycles in atomic operations and store buffer stalls)
    - but, copy-back and invalidate event counts agreed much more closely
  - inaccuracies in D-cache probably due to random replacement policy

15 Performance: Slowdown Incurred by Detailed Modelling

- bridge-based memory system structure (non-pipelined backplane) introduced a 25%–15% overhead (over 1–8 threads) over original structure
- further slowdown for full pipelined modelling for the NPB (S class): (new)

threads	BT	CG	FT	IS	LU	LU-hp	MG	SP
1	1.19	1.02	1.27	1.09	1.10	1.12	1.06	1.15
2	1.22	1.08	1.32	1.19	1.16	1.32	1.10	1.19
4	1.23	1.12	1.36	1.11	1.17	1.52	1.27	1.31
8	1.26	1.28	1.41	1.25	1.17	1.65	1.37	1.43

- performance impact varied with level of modelling detail
  - P-cache and store buffer modelling introduced bulk of overhead (where used intensively)
  - overhead of pipelined backplane itself less than 20%
- non-pipelined model was more optimistic on NPB: 0-10% (1 thread) and 10-20% (4 threads)

(↑ with degree of data sharing)

16 Conclusions

- accurate simulation of a modern NUMA memory system achieved by:
  - bridge-based structure
  - event windows (PDES) for efficient modelling of asynchronous events
  - processor model only slightly extended (event queue and stall exceptions)
  - overhead within a factor of two, generally much less
  - store buffer and P-cache most problematic (complexity, overhead and accuracy)
- 2-stage validation methodology was reasonably successful
  - microbenchmarks for basic calibration (+ discovery of undocumented effects)
  - OpenMP NPB, augmented with hardware event counting, to pinpoint more subtle areas of inaccuracy
  - level of accuracy fair, simulator tending to be slightly optimistic
  - entails hard work, and is limited by lack of accurate and complete documentation of the target system

17 Future Work and Acknowledgements

- future work:
  - further refine validation
    - isolate differences in (atomic-operation rich) barriers
    - discrepancies here may be polluting many event counts
  - parallelise processor simulation (can analyse larger workloads)
  - model contention on the data bus and NUMA effects
  - explore impact of variations in simulated memory system design
- acknowledgements:
  - CC-NUMA Project
  - Australian Research Council
  - Sun Microsystems
  - Gaussian Inc
  - MASCOTS!