Multi-Execution: Multicore Caching for Data-Similar Executions

Susmit Biswas, Diana Franklin, Alan Savage, Ryan Dixon, Timothy Sherwood, Frederic T Chong

susmit@cs.ucsb.edu
How to use so many cores?

Transactional memory [Herlihy-ISCA93]
Thread level speculation [Steffan-ISCA00]
...
How to use so many cores?

- “Multi-execution”
- Run multiple instances
  - Serial code
  - Input variation

Sic3a Analog Circuit Simulator

susmit@cs.ucsb.edu
Multi-Execution: Is it common?

- Simulation
- CAD
- Machine Learning
- Data Hiding
Multi-Execution Data Similarity

1 MB Direct Mapped Cache

Process 1

1 MB Direct Mapped Cache

Process 2

Input 1

Application

Input 2

susmit@cs.ucsb.edu
Multi-Execution Data Similarity

1 MB Direct Mapped Cache

255.vortex
188.ammp
175.vpr
300.twolf

% Similarity of caches

References (Million)
Data Block Merging

Shared Memory

Processors

P₀

P₁

P₂

P₃

susmit@cs.ucsb.edu
Effect of Cache Capacity on Miss Rate

- # L2 Miss / 1K memory refs
- Cache size (Megabyte)

- 22.3
- 7.8

susmit@cs.ucsb.edu
Challenges

• Fast searching of identical blocks
  – Similarity at same virtual address
  – Limit within set

• Track Merged blocks
  – Bit per processor in tag

• Unmerge
  – Exclusive policy
  – unmerge on L1 miss
Modification in Addressing

- 32 bit addr
- 4MB, 8W
- 14 bits index
- 4KB page
- 16 Proc

Lower $\log_2(p)$-bits of page number, $p=$number of processors

susmit@cs.ucsb.edu
Mergeable Cache Design

Extra 5.28% power, 4.21% area of 4MB cache

susmit@cs.ucsb.edu
Simulation Framework

- Polyscalar OOO-multiprocessor simulator
- 2 – 8 instances of applications

- L1 cache: 32KB-I+32KB-D direct mapped
- L1 latency: 1 cycle

- Shared L2 cache: 4MB-8way-32B lines
- L2 latency: 6 cycles

- DRAM latency 200 cycles
## Benchmarks

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Description</th>
<th>Input Modification</th>
<th>Run Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>175.vpr</td>
<td>FPGA Place and Route</td>
<td>routing-channel-width</td>
<td>3.3 B</td>
</tr>
<tr>
<td>181.mcf</td>
<td>Combinatorial Optimization</td>
<td>reduced trips</td>
<td>6.99 B</td>
</tr>
<tr>
<td>183.equake</td>
<td>Seismic Wave Propagation</td>
<td>epicenter and intensity</td>
<td>4.36 B</td>
</tr>
<tr>
<td>188.ammp</td>
<td>Chemistry</td>
<td>simulation parameters</td>
<td>6.13 B</td>
</tr>
<tr>
<td>255.vortex</td>
<td>Database</td>
<td>random insert, lookup</td>
<td>5.85 B</td>
</tr>
<tr>
<td>300.twolf</td>
<td>Place and Route</td>
<td>intercell gaps</td>
<td>4.13 B</td>
</tr>
<tr>
<td>libsvm</td>
<td>Machine learning</td>
<td>$C$ and $\gamma$ parameter</td>
<td>5.67 B</td>
</tr>
<tr>
<td>icsiboost</td>
<td>Ensemble learning</td>
<td>distribution of sample</td>
<td>2.30 B</td>
</tr>
</tbody>
</table>
Results: Off-chip Accesses

Lower is better

% L2 Misses

icsboost  |  libsvm  |  twolf  |  equake  |  ammp  |  mcf  |  vpr  |  vortex

2 Processes  |  4 Processes  |  8 Processes

Poor scaling

Good Scaling

Little Worse
Improvement Over No Merging

Higher is better

6.92X

Little Worse

icsiboost libsvm twolf ammp equake vpr mcf vortex

2 Processes
4 Processes
8 Processes

susmit@cs.ucsb.edu
Conclusion

• “Multi-execution” domain

• Data similarity exists

• Mergeable cache
  – improves performance
    • 6.92X max, 2.5X in average
Future Work

• Dynamic hybrid mergeable & conventional cache

• Wider application set
  – More applications
  – Libraries

• Coarse-grain software implementations
Thanks!

http://www.cs.ucsb.edu/~arch

susmit@cs.ucsb.edu