CHIP MULTIPROCESSORS

Power Dissipation Limits Practical Clock Frequency

Historical Processor Performance

Chip Multiprocessors (CMPs)

Outline

- Chip Multiprocessors (CMPs)
- CMP Resource Management
- Miss Bandwidth Management
  - Greedy Miss Bandwidth Management
  - Interference Measurement
  - Model-Based Miss Bandwidth Management
- Off-Chip Bandwidth Management
- Conclusion
Projected Number of Cores

Observation 1: Multiprogramming can provide near-term throughput improvement.

Processor Memory Gap

Observation 3: Latency hiding techniques are necessary.

Performance vs. Bandwidth

Observation 4: Bandwidth must be used efficiently.

Application Trends
- Multi-programming
- Software parallelism
- Latency hiding
- Bandwidth efficiency

Hardware Trends
- Concurrent applications share hardware
- Complex Memory Systems

Shared Resource Management

Why Manage Shared Resources?
- Provide predictable performance
- Support OS scheduler assumptions
- Cloud: Fulfill Service Level Agreement

CMP RESOURCE MANAGEMENT
Performance Variability Metrics

- **Fairness**
  - The performance reduction due to interference between processes is distributed across all processes in proportion to their priorities.
  - Equal priorities: Performance reduction from sharing affects all processes equally.

- **Quality of Service**
  - The performance of a process is never drops below a certain limit regardless of the behavior of co-scheduled processes.

### Performance Variability (Fairness)

![Graph showing performance variability (fairness) across different workloads and network architectures.]

<table>
<thead>
<tr>
<th>Number of Workloads</th>
<th>Lowest Fairness Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>0.4</td>
</tr>
<tr>
<td>5</td>
<td>0.6</td>
</tr>
<tr>
<td>7</td>
<td>0.8</td>
</tr>
<tr>
<td>9</td>
<td>1.0</td>
</tr>
<tr>
<td>11</td>
<td>1.2</td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
</tr>
</tbody>
</table>

Resource Management Tasks

- **Measurement**
- **Allocation (Policy)**
- **Enforcement (Mechanism)**

### Contributions

- Offline Interference Measurement
- Dynamic Miss Handling Architecture
- Greedy Miss Bandwidth Allocation
- Performance Model-Based Bandwidth Allocation
- Low-Cost Open Page Prefetching
- Opportunistic Prefetch Scheduling

Conventional Resource Allocation Implementation

- Measurement
- Allocation
- Enforcement

**Miss Bandwidth Management**

**GREEDY MISS BANDWIDTH MANAGEMENT**
Alternative Resource Allocation Implementation

Dynamic Miss Handling Architecture

A DMHA controls the number of concurrent shared memory system requests that are allowed for each processor.

Greedy Miss Bandwidth Management

- Idea: Reduce the number of MSHRs if a metric exceeds a certain threshold
- Metrics:
  - Paper A.II: Memory bus utilization
  - Paper A.III: Simple interference counters (Interference Points)
- Performance feedback avoids excessive performance degradations

Resource Allocation Baselines

Baseline = Interference-free configuration

Quantify performance impact from interference

Private Mode and Shared Mode

Interference Definition

\[ I_i = \hat{L}_i - L_i \]

\[ E_i = \hat{L}_i - \hat{L}_i \]
Offline Interference Measurement

Interference Penalty Frequency (IPF) counts the number of requests that experienced an interference latency of \( i \) cycles.

Interference Impact Factor (IIF) is the interference latency times the probability of it arising, i.e. \( IIF(i) = i \times P(i) \).

Aggregate Interference Impact

Resource Management Baselines

Baseline Weaknesses

Multiprogrammed Baseline
- Only accounts for interference in partitioned resources
- Static and equal division of DRAM bandwidth does not give equal latency
- Complex relationship between resource allocation and performance

Single Program Baseline
- Does not exist in shared mode

Online Interference Measurement

- Dynamic Interference Estimation Framework (DIEF)
- Estimates private mode average memory latency
- General, component-based framework

Shared Cache Interference

Miss

Eviction may not be interference

Eviction is interference

Interference latency cost = miss penalty
Bus Interference Requirements

- Out-of-order memory bus scheduling
- Shared mode only cache misses and cache hits
- Shared cache writebacks

Computing private latency based on shared mode queue contents is difficult

Emulate private scheduling in the shared mode

Model-Based Miss Bandwidth Allocation

DIEF provides accurate estimates of the average private mode memory latency

Can we use the estimates provided by DIEF to choose miss bandwidth allocations?

We need a model that relates average memory latency to performance

Performance Model

Observation: The memory latency performance impact depends on the parallelism of memory requests

Very similar in private and shared mode

Bandwidth Management Flow

Measurement

- Shared Mode Memory Latency
- Private Mode Memory Latency
- CPU Stall Time
- Committed Instructions
- Number of Memory Requests

Modeling

- Per-CPU Models
- Perf Metric Model

Allocation

Find MSHR allocation that maximizes the chosen performance metric

Set number of MSHRs for all last-level private caches
OFF-CHIP BANDWIDTH MANAGEMENT

Modern DRAM Interfaces
- Maximize bandwidth with 3D organization
- Repeated requests to the row buffer are very efficient

Low-Cost Open Page Prefetching
- Idea: Piggyback prefetches to open DRAM pages on demand reads
- Performance win if prefetcher accuracy is above ~40%

Opportunistic Prefetch Scheduling
- Idea: Issue prefetches when a page is closed
- Increased efficiency: 8 transfers for 3 activations

Conclusion
- Managing bandwidth allocations can improve CMP system performance
- Miss bandwidth management
  - Greedy allocations
  - Management guided by accurate measurements and performance models
- Off-chip bandwidth management with prefetching

CONCLUSION
Thank You

Visit our website:
http://research.idi.ntnu.no/multicore/

EXTRA SLIDES

Future Work

- Performance-directed management of shared caches and the memory bus
- Improving OS and system software with dynamic measurements
- Combining dynamic MHAs with prefetching to improve system performance
- Managing workloads of single-threaded and multi-threaded benchmarks

Example Chip Multiprocessor