The Memory Hierarchy

This work supported by UNSW and HP through the Gelato Federation

Why a memory hierarchy?:
- Fast vs Expensive
- Exploit locality
  - Spatial
  - Temporal

Cache Levels
- L1/L2 usually on chip
- L2 usually unified
- L3 much larger
- L1 tied to clock rate, lower levels tied to miss cost of L1

Cache Lines
- Split cache
  - Instruction cache
  - Data cache

Cache Design

Cache Organisation

Where can a cache line live?:
- Direct Mapped
- Fully Associative
- Set Associative
  - n-way set associative is to divide total cache into n compartments.
  - Line may live anywhere in set total blocks mod n

How do we find a cache line?:
- Index
- Tag
  - \( \text{index bits} = \log_2(\text{total blocks}) \)
WHERE IS THAT LINE - TAGS AND INDEXES

More Associative

Virtual Address

INDEX

TAG

More set-associativity means more tag bits

Offset

Way 1

Way 2

Way 3

Way 4

Virtual Address

INDEX

TAG

Less Associative

More set-associativity means more index bits

Offset

Way 1

Way 2

Way 3

Way 4

Quickly - Other Cache Parameters

- Replacement Policy
  - LRU/Random/FIFO
- Write policy
  - Through
  - Back
- Inclusive or Exclusive?

Power

CPU

L2

Interconnect

L3

Memory Controller

RAM

Power4

CPU

L2

Interconnect

L3

Memory Controller

RAM

Power5

CPU

L2

Interconnect

L3

Memory Controller

RAM

Chronology of a Cache Hit

1. Processor loads from an address
2. Address is requested in the cache
3. Index selects offset within (all) ways
4. Tag selects correct entry from set
5. Data is retrieved from selected line

Cache addressing: This address is a virtual address

- Virtual addresses may alias
- Cache must be coherent
- Synonyms : ∆ VA, ≡ page
- Homonyms : ≡ page, ∆ VA
**Physically Indexed and Tagged Cache**

- TLB translates VA to PA
- No aliases
- Extra overhead

**Virtually Tagged, Virtually Indexed**

- TLB less involved
- No size limitations
- Synonyms and homonyms

**Protection**

- TLB still required for protection
- Active research area
  - Protection details in cache
  - PLB
  - Capabilities
  - Segmentation

**Physically Tagged, Virtually Indexed**

- Tag is based on PA
- Index based on untranslated offset bits
- TLB lookup happens in parallel
- Index limited to system page size
- Unless bits are shared...
- ...which introduces aliasing

**Dealing with Aliasing**

- Flush cache on context switch (Sledgehammer)

**Software Approaches**

- SASOS
- Mungi
- Colouring
  - Make shared items align in the cache (SunOS)
  - Globally visible shared region (OS/2)
Dealing with Aliasing

Hardware Approaches:
- Reverse Maps
- Back Pointers (MIPS R6000, Alpha?)
- Dual Directories
- ASID or segmentation
  - Add bits to distinguish VAs
  - Makes sharing harder
  - Itanium Region Registers

Dealing with misses

So far, everything has been about hit latency

Types of misses:
1. Compulsory
2. Capacity
3. Conflict

More cache?
- Less compulsory misses
- $$$
- \( \text{cache size} = \text{line size} \times \text{set index} \times \text{associativity} \)
- \( \uparrow \text{cache} \) means more what?
  - Greater line size
  - Greater associativity
  - Greater index size
### Other Miss Penalty Reduction Schemes

- Critical word first
- Victim caches
- Way prediction
- Trace Cache

### Prefetching

- Requires non-blocking cache

### Prefetching Example

Walk an array in cache sized lines

<table>
<thead>
<tr>
<th>Metric</th>
<th>ICC</th>
<th>GCC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run Time (seconds)</td>
<td>0.824</td>
<td>1.507</td>
<td>Program execution time</td>
</tr>
<tr>
<td>L1 Data Cache Hit Rate</td>
<td>99.96%</td>
<td>99.99%</td>
<td>L1 Data Cache Hit Rate</td>
</tr>
<tr>
<td>L1 Data Cache Misses</td>
<td>12,514,832</td>
<td>12,505,200</td>
<td>L1 Data Cache Misses</td>
</tr>
<tr>
<td>Full Pipe Bubbles in Main Pipe due to Execution Unit Stalls</td>
<td>129,629,838</td>
<td>737,931,717</td>
<td>Full Pipe Bubbles in Main Pipe due to Execution Unit Stalls</td>
</tr>
<tr>
<td>Back-end was stalled by issue due to GR/GR dependency</td>
<td>0</td>
<td>0</td>
<td>Back-end was stalled by issue due to GR/GR dependency</td>
</tr>
</tbody>
</table>

### IF YOU ARE STILL AWAKE, I OWE YOU A BEER

Questions?