CACHE ARCHITECTURE

Mahdi Nazm Bojnordi

Assistant Professor
School of Computing
University of Utah
Overview

- Announcement
  - Homework 4 will be released on Oct. 30th

- This lecture
  - Cache addressing and lookup
  - Cache optimizations
    - Techniques to improve miss rate
    - Replacement policies
    - Write policies
Recall: Cache Optimizations

- How to improve cache performance?

  \[ AMAT = t_h + r_m \cdot t_p \]

- Reduce hit time \((t_h)\)

- Improve hit rate \((1 - r_m)\)

- Reduce miss penalty \((t_p)\)
Recall: Cache Optimizations

- How to improve cache performance?
  \[ AMAT = t_h + r_m t_p \]

- Reduce hit time \((t_h)\)
  - Memory technology, critical access path

- Improve hit rate \((1 - r_m)\)

- Reduce miss penalty \((t_p)\)
Recall: Cache Optimizations

- How to improve cache performance?
  \[ AMAT = t_h + r_m t_p \]

- Reduce hit time \((t_h)\)
  - Memory technology, critical access path

- Improve hit rate \((1 - r_m)\)
  - Size, associativity, placement/replacement policies

- Reduce miss penalty \((t_p)\)
Recall: Cache Optimizations

- How to improve cache performance?
  \[ AMAT = t_h + r_m t_p \]

- Reduce hit time \((t_h)\)
  - Memory technology, critical access path

- Improve hit rate \((1 - r_m)\)
  - Size, associativity, placement/replacement policies

- Reduce miss penalty \((t_p)\)
  - Multi level caches, data prefetching
Set Associative Caches

- Improve cache hit rate by allowing a memory location to be placed in more than one cache block
  - N-way set associative cache
  - Fully associative
- For fixed capacity, higher associativity typically leads to higher hit rates
  - more places to simultaneously map cache lines
  - 8-way SA close to FA in practice

```c
for (i=0; i<10000; i++) {
    a++;
    b++;
}
```
Set Associative Caches

- Improve cache hit rate by allowing a memory location to be placed in more than one cache block
  - N-way set associative cache
  - Fully associative
- For fixed capacity, higher associativity typically leads to higher hit rates
  - more places to simultaneously map cache lines
  - 8-way SA close to FA in practice

for (i=0; i<10000; i++) {
  a++;
  b++;
}

```
…
```
n-Way Set Associative Lookup

- Index into cache sets
- Multiple tag comparisons
- Multiple data reads
- Special cases
  - Direct mapped
    - Single block sets
  - Fully associative
    - Single set cache

```
<table>
<thead>
<tr>
<th>tag</th>
<th>index</th>
<th>byte</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
v
0
1
...
510
511
```

```
= =
mux
```

```
data
hit
OR
```
Example Problem

- Find the size of tag, index, and offset bits for an 4MB, 4-way set associative cache with 32B cache blocks. Assume that the processor can address up to 4GB of main memory.
Example Problem

- Find the size of tag, index, and offset bits for an 4MB, 4-way set associative cache with 32B cache blocks. Assume that the processor can address up to 4GB of main memory.

- $4\text{GB} = 2^{32} \text{B} \Rightarrow \text{address bits} = 32$
- $32\text{B} = 2^5 \text{B} \Rightarrow \text{byte offset bits} = 5$
- $4\text{MB}/(4\times32\text{B}) = 2^{15} \Rightarrow \text{index bits} = 15$
- $\text{tag bits} = 32 - 5 - 15 = 12$
Cache Miss Classifications

- Start by measuring miss rate with an ideal cache
  - 1. ideal is fully associative and infinite capacity
  - 2. then reduce capacity to size of interest
  - 3. then reduce associativity to degree of interest
Cache Miss Classifications

- Start by measuring miss rate with an ideal cache
  - 1. ideal is fully associative and infinite capacity
  - 2. then reduce capacity to size of interest
  - 3. then reduce associativity to degree of interest

1. Cold (compulsory)
Cache Miss Classifications

- Start by measuring miss rate with an ideal cache
  - 1. ideal is fully associative and infinite capacity
  - 2. then reduce capacity to size of interest
  - 3. then reduce associativity to degree of interest

1. Cold (compulsory)

- Cold start: first access to block
- How to improve
  - large blocks
  - prefetching
Cache Miss Classifications

- Start by measuring miss rate with an ideal cache
  - 1. ideal is fully associative and infinite capacity
  - 2. then reduce capacity to size of interest
  - 3. then reduce associativity to degree of interest

1. Cold (compulsory)
2. Capacity

- Cold start: first access to block
- How to improve
  - large blocks
  - prefetching
Cache Miss Classifications

- Start by measuring miss rate with an ideal cache
  - 1. ideal is fully associative and infinite capacity
  - 2. then reduce capacity to size of interest
  - 3. then reduce associativity to degree of interest

1. Cold (compulsory)
   - Cold start: first access to block
   - How to improve
     - large blocks
     - prefetching

2. Capacity
   - Cache is smaller than the program data
   - How to improve
     - large cache
Cache Miss Classifications

- Start by measuring miss rate with an ideal cache
  - 1. ideal is fully associative and infinite capacity
  - 2. then reduce capacity to size of interest
  - 3. then reduce associativity to degree of interest

1. Cold (compulsory)
   - Cold start: first access to block
   - How to improve:
     - large blocks
     - prefetching

2. Capacity
   - Cache is smaller than the program data
   - How to improve:
     - large cache

3. Conflict
Cache Miss Classifications

- Start by measuring miss rate with an ideal cache
  - 1. ideal is fully associative and infinite capacity
  - 2. then reduce capacity to size of interest
  - 3. then reduce associativity to degree of interest

1. Cold (compulsory)
   - Cold start: first access to block
   - How to improve
     - large blocks
     - prefetching

2. Capacity
   - Cache is smaller than the program data
   - How to improve
     - large cache

3. Conflict
   - Set size is smaller than mapped mem. locations
   - How to improve
     - large cache
     - more assoc.
100,000 loads and stores are generated; L1 cache has 3,000 misses; L2 cache has 1,500 misses. What are various miss rates?
Miss Rates: Example Problem

- 100,000 loads and stores are generated; L1 cache has 3,000 misses; L2 cache has 1,500 misses. What are various miss rates?

- L1 miss rates
  - Local/global: 3,000/100,000 = 3%

- L2 miss rates
  - Local: 1,500/3,000 = 50%
  - Global: 1,500/100,000 = 1.5%
Cache Replacement Policies

- Which block to replace on a miss?
  - Only one candidate in direct-mapped cache
  - Multiple candidates in set/fully associative cache
Cache Replacement Policies

- Which block to replace on a miss?
  - Only one candidate in direct-mapped cache
  - Multiple candidates in set/fully associative cache

- Ideal replacement (Belady’s algorithm)
  - Replace the block accessed farthest in the future

<table>
<thead>
<tr>
<th>Cache Set</th>
<th>Requested Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>--</td>
<td>A B C B B B C A</td>
</tr>
<tr>
<td>--</td>
<td></td>
</tr>
</tbody>
</table>
Cache Replacement Policies

- Which block to replace on a miss?
  - Only one candidate in direct-mapped cache
  - Multiple candidates in set/fully associative cache

- Ideal replacement (Belady’s algorithm)
  - Replace the block accessed farthest in the future

<table>
<thead>
<tr>
<th>Cache Set</th>
<th>Requested Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>C</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
</tr>
</tbody>
</table>