Page Tables Revisited

Two-level Translation

- Can be optimised for TLB refill only
  - Does not need to check the exception type
  - Does not need to save any registers
    - It uses a specialised assembly routine that only uses k0 and k1.
    - Does not check if PTE exists
- It uses a virtual linear array – see extended OS notes
- With careful data structure choice, exception handler can be made very fast

- An example routine
  mfc0 k1,0
  mfc0 k0,0
  lw k1,0(k1) # may double fault (k0 = orig EPC)
  nop
  mfc0 k1,0
  mtc0 k1,0
  tlbwr
  jr k0
  rfe

Virtual Linear Array page table

- Assume a 2-level PT
- Assume 2nd-level PT nodes are in virtual memory
- Assume all 2nd-level nodes are allocated contiguously ⇒ 2nd-level nodes form a contiguous array indexed by page number

Virtual Linear Array Operation

- Index into 2nd level page table without referring to root PT!
- Simply use the full page number as the PT index!
- Leave unused parts of PT unmapped!
- If access is attempted to unmapped part of PT, a secondary page fault is triggered
  - This will load the mapping for the PT from the root PT
  - Root PT is kept in physical memory (cannot trigger page faults)
Virtual Linear Array Page Table
• Use Context register to simply load PTE by indexing a PTE array in virtual memory
• Occasionally, will get double faults
  – A TLB miss, while servicing a TLB miss
  – Handled by general exception handler
  
  • Protected from user access

Code for VLA TLB refill handler

Software-loaded TLB
• Pros
  – Can simplify hardware design
  – Provide greater flexibility in page table structure
• Cons
  – Typically have slower refill times than hardware managed TLBs.

Trends
• Operating systems
  – Moving functionality into user processes
  – Making greater use of virtual memory for mapping data structures held within the kernel.
• RAM is increasing
  – TLB capacity is relatively static
• Statement:
  – Trends place greater stress upon the TLB by increasing miss rates and hence, decreasing overall system performance.
  – True/False? How to evaluate?
Note the TLB miss costs

- What is expected to be the common case?
Specialising the L2/L1K miss vector

<table>
<thead>
<tr>
<th>Type of PTE Miss</th>
<th>Counts</th>
<th>Previous Total Cost (Table 6)</th>
<th>New Total Cost (Table 6)</th>
<th>Time Saved (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match-Coalesced</td>
<td>30,120,723</td>
<td>30.15</td>
<td>30.15</td>
<td>0.00</td>
</tr>
<tr>
<td>L2</td>
<td>5,000,894</td>
<td>8.99</td>
<td>6.79</td>
<td>2.20</td>
</tr>
<tr>
<td>L1K</td>
<td>2,493,282</td>
<td>43.85</td>
<td>2.89</td>
<td>40.96</td>
</tr>
<tr>
<td>L3</td>
<td>999,431</td>
<td>11.85</td>
<td>11.85</td>
<td>0.00</td>
</tr>
<tr>
<td>Identity</td>
<td>127,245</td>
<td>3.81</td>
<td>3.81</td>
<td>0.00</td>
</tr>
<tr>
<td>Invalid</td>
<td>198,439</td>
<td>2.10</td>
<td>2.10</td>
<td>0.00</td>
</tr>
<tr>
<td>Total</td>
<td>53,033,115</td>
<td>106.56</td>
<td>68.25</td>
<td>48.31</td>
</tr>
</tbody>
</table>

Table 7: Recomputed Cost of TLB Misses Given Additional Miss Vectors (Mach 3.0)

Other performance improvements?

- In Paper
  - Pinned slots
  - Increased TLB size
  - TLB associativity
- Other options
  - Bigger page sizes
  - Multiple page sizes

Itanium Page Table

- Takes a bet each way
- Loading
  - software
  - two different format hardware walkers
- Page table
  - software defined
  - linear
  - hashed

Measurement Results

<table>
<thead>
<tr>
<th>System</th>
<th>LRU</th>
<th>L1C</th>
<th>L2</th>
<th>L3</th>
<th>Invalid</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mach 3.0</td>
<td>560</td>
<td>6.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mach 3.0</td>
<td>560</td>
<td>6.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mach 3.0</td>
<td>560</td>
<td>6.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mach 3.0</td>
<td>560</td>
<td>6.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 5: Number of TLB Misses

<table>
<thead>
<tr>
<th>System</th>
<th>Total TLB Service Time (ms)</th>
<th>LRU</th>
<th>L1C</th>
<th>L2</th>
<th>L3</th>
<th>Invalid</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mach 3.0</td>
<td>560</td>
<td>6.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mach 3.0</td>
<td>560</td>
<td>6.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mach 3.0</td>
<td>560</td>
<td>6.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mach 3.0</td>
<td>560</td>
<td>6.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 6: Time Spent Handling TLB Misses