Efficient Memory Virtualization
Reducing Dimensionality of Nested Page Walks
Jayneel Gandhi†, Arkaprava Basu‡, Mark D. Hill†, Michael M. Swift†

1. Problem
Hardware Virtualized MMU have high overheads
We will show that the increase in translation lookaside buffer (TLB) miss handling costs due to the hardware-assisted memory management unit (MMU) is the largest contributor to the performance gap between native and virtual servers.

2. Why is a TLB miss costlier?

3. Solution
Segmentation to bypass paging

- Extend Direct Segments for virtualization
- Direct Segment at VMM, guest or both levels
- Three modes with different tradeoffs

4. Configurations

5. Tradeoffs

<table>
<thead>
<tr>
<th>Properties</th>
<th>Base Virtualized</th>
<th>Dual Direct (new)</th>
<th>VMM Direct (new)</th>
<th>Guest Direct (new)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page walk dimensions</td>
<td>2D</td>
<td>0D</td>
<td>1D</td>
<td>1D</td>
</tr>
<tr>
<td># of memory accesses for most page walks</td>
<td>24</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td># of base-bound checks for page walks</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Guest OS modifications</td>
<td>none</td>
<td>required</td>
<td>none</td>
<td>required</td>
</tr>
<tr>
<td>VMM modifications</td>
<td>none</td>
<td>required</td>
<td>required</td>
<td>none</td>
</tr>
<tr>
<td>Application category</td>
<td>any</td>
<td>big memory</td>
<td>any</td>
<td>big memory</td>
</tr>
<tr>
<td>Page sharing</td>
<td>unrestricted</td>
<td>limited</td>
<td>limited</td>
<td>unrestricted</td>
</tr>
<tr>
<td>ballooning</td>
<td>unrestricted</td>
<td>limited</td>
<td>limited</td>
<td>unrestricted</td>
</tr>
<tr>
<td>Guest OS swapping</td>
<td>unrestricted</td>
<td>limited</td>
<td>unrestricted</td>
<td>limited</td>
</tr>
<tr>
<td>VMM swapping</td>
<td>unrestricted</td>
<td>limited</td>
<td>limited</td>
<td>unrestricted</td>
</tr>
</tbody>
</table>

6. Optimizations
- Guest physical memory fragmentation: **Self-ballooning**
  - Balloon-out fragmented memory and provide to VMM
  - VMM hot-adds new contiguous guest physical memory
- Host physical memory fragmentation: **Compaction**
  - Remap fragmented pages to create contiguous physical memory
- Permanent “hard” memory faults: **Escape filter**
  - Escape filter stores few pages with permanent “hard” faults
  - Escape filter checked in parallel with VMM segment register
  - If found in escape filter, get alternate translation through paging

7. Overheads + Results
Near- or better-than-native performance

---

Acknowledgement
This work is supported in part by the National Science Foundation (CNS-1117280, CCF-1218323, CNS-1302260 and CCF-1438992), Google, and the University of Wisconsin (Kellett award and Named professorship to Hill). Arkaprava Basu’s contribution to this work occurred while at UW-Madison.