

1 Summary

Efficient Virtual Memory for Big Memory Servers (2013): The authors propose an efficient memory mapping technique called direct-segment. As memory resources increase and the cost decreases, applications are trending towards larger memory requirements. This leads to the increase in overhead of address translation and therefore the cost of TLB misses. Big-memory workloads incur substantial performance loss in using page-based virtual memory with TLB misses, as they can waste up to 51% of execution cycles on TLB misses and translations. In general, most of the memory usage here doesn't benefit from features of paged virtual memory and still suffers the virtual memory cost. The authors propose, in addition to default page mappings, mapping parts of the virtual address space with direct segments. A large range of contiguous virtual memory addresses maps to contiguous physical memory addresses using just a few additional registers (base, limit, and offset) per core; addresses in this particular range (called primary region) get translated without the possibility of a TLB miss, while those outside the range are page mapped and translated using TLB and the normal paging mechanisms. The authors show using an emulated direct-segment memory mapping hardware and software support for it, that direct-segment can reduce that time wasted to less than 0.5% of execution cycles for big-memory workloads, eliminating almost all TLB performance loss for long-running, big-memory workloads.

2 Strengths

- The paper's characterization of big-memory workloads is in-depth and it makes reasonable explanations of why certain paging features aren't required for most of memory in these workloads with evidence. Swapping isn't useful because performance-critical applications cannot afford to wait on disk I/O, fragmentation mitigation is not required because most of memory is allocated early in execution and memory usage stays predictable, and fine-grain per-page protection is not significant because most of memory get the same read-write permissions.

3 Weaknesses

- Direct-segment doesn't sound necessarily new or original. The small hardware implementation for it sounds a lot like how base-and-bounds (dynamic relocation) works, but built to work alongside paging and TLBs, which both came after dynamic relocation.
- The paper focuses on addressing problems with paging in big-memory workloads and how they can benefit from direct-segment. For breadth, it would be good to include how less-memory-intensive workloads perform using direct-segment, which could further highlight the different needs (or lack thereof) of other applications in terms of virtualized memory implementations. This is only briefly discussed in Section 3.3 on the limitations of their solution.

4 Rating: 4

5 Comments

While it seems like the authors did not set out make a point about domain specific architectures (DSAs), this work still highlights the benefit and need for these application specific designs that improve on existing ones. The authors state that their goal is to address an issue in TLB performance and that they think future work in direct-segments (or similar) could become a fully general solution for TLB performance (Section 3.3). However, the paper touches on the different memory resource needs and usage patterns seen specifically in big-memory servers, which isn't fully served by how virtual memory was implemented with paging at the time. To fully understand the performance needs of big-memory applications, they analyze those particular workloads, pointing out where paging fell short and implementing a mechanism, direct-segment, to fill the gaps. Direct-segment in this paper isn't something that could fully replace paged virtual memory because it is not fully general, but the methodology to design exploits that the authors did in the paper appears especially significant for interested enterprise-scale applications that rely heavily on performance. It is fairly well understood that with the large growth and diverse range of how technology is used, that domain or application specific technology is growing in significance. Since the 60s, it seems that page-based virtual memory and TLBs are the default in terms of general memory management systems because it is versatile across many architectures. General-purpose architectures tend to perform sufficiently, but designers can still extract more performance gains by building for domain specific needs when general-purpose solutions are difficult, which this paper indirectly addresses before DSAs became a more widespread and popular topic.