

As systems provide increasing memory capacities to support memory-intensive workloads, Translation Lookaside Buffers (TLBs) are becoming a critical performance bottleneck. TLB performance is exacerbated with virtualization, which is typically implemented with two-dimensional nested page tables. While virtualization vendors recommend using large pages to mitigate TLB overheads, in practice, page splintering (where the guest utilizes large pages, only to be broken by the hypervisor, or vice-versa) is prevalent and severely restricts large page benefits. We characterize real-system page allocation behavior under virtualization and find that the guest operating system (OS) often aggressively allocates large pages, which are then broken by the hypervisor to baseline pages. However, in trying to allocate large pages, the hypervisor usually allocates these baseline system physical pages in aligned, contiguous frames with respect to the application's guest virtual address. In response, we propose hardware to, on TLB misses, speculate on the system physical address based on the application's virtual address. TLB speculation can be readily overlaid on various TLB organizations and is effective under a wide range of real-world virtual machine deployments. Overall, TLB speculation eliminates almost all of the TLB handling overheads of virtualization.