Shashi Shivaraju
XID:C88650674

Paper Review 3- Practical, transparent operating system support for superpages

This paper describes the various challenges with OS-level support for super-pages and details the design of an effective super-pages management system. A superpage is a memory page of larger size than an hardware base page. Modern general-purpose processors provide virtual memory support, using page tables for address translation. Most processors cache virtual-to-physical-address mappings from the page tables in a translation look aside buffer (TLB). TLB coverage is defined as the amount of memory accessible through these cached mappings, i.e., without incurring misses in the TLB. Reduced TLB coverage makes virtual memory systems less efficient. To alleviate this problem, most modern general-purpose CPUs provide support for superpages.

In super-pages approach, hardware supports more than one page size and the super-page size is a power-of-2 multiple of the base page size. Thus super-page maps several base pages and thus reduce the size of the page table, increase TLB coverage and optimizes I/O time. But in-turn the use of multiple page sizes leads to the problem of physical memory fragmentation, and decreases future opportunities for using large superpages. To ensure sustained performance, the operating system needs to control fragmentation, without penalizing system performance. The problem of effectively managing superpages thus becomes a complex, multi-dimensional optimization task. Most general-purpose operating systems either do not support superpages at all, or provide limited support. This paper develops a general and transparent superpage management system.

The paper details the hardware constraints, issues and tradeoffs in operating system support for superpages. Hardware constraints on super-pages are that there must be enough contiguous free memory to store each superpage,hardware might support limited page sizes and each superpage addresses must be aligned on the superpage size. TLB entry for a superpage provides only a single reference bit, dirty bit, and set of protection attributes. Due to the coarse granularity of reference and dirty bits, the operating system can determine whether some part of the superpage has been accessed or written to, but cannot distinguish between base pages in this regard.

The paper details the design decision on super page allocation wherein we acquire base pages on demand and “promote” to superpage at some later date or load entire superpage when when one base page is faulted in. In acquire on demand and promote approach there is Reservation-based allocation wherein we set aside space for a superpage when first base page is loaded or Relocation-based allocation wherein we wait until a superpage is formed and then move existing pages to contiguous locations. The authors of the paper have developed a reservation-based system in which a page is initially loaded choose a superpage size and reserve contiguous frames to hold the eventual superpage.

The paper details the design decision on promotion,demotion and eviction of super-pages. Promotion should occur when enough pages have been loaded to justify creating a superpage and then TLB entries are combined into one entry with page table updated to show new superpage size. Here early promotion is a trade-off since it reduces TLB misses but wastes memory if all pages of the superpage are not required. Demotion is used to reduces superpage into individual base pages or to a smaller superpage. Demotion is required if memory is needed for new pages by evicting unused base pages. Eviction process on a superpage can occur when memory is full and results in release of all base pages of the superpage.

Shashi Shivaraju
XID:C88650674

The authors of the paper have implemented a reservation-based superpage management system in FreeBSD on the Alpha CPU, and evaluated it on real workloads and benchmarks. The system uses preemption to preempt an existing reservation that has enough unallocated frames to satisfy the request and uses it whenever possible. It supports multiple superpage sizes to reduce internal fragmentation and uses “buddy allocator” to reduce fragmentation. Also a page replacement daemon is used periodically which selects pages to be swapped out and it is modified to include contiguity as one of the factors to be considered. This system will promote sub-superpage only its regions that are fully populated. It demotes infrequently referenced pages to reclaim memory frame. When a base page is evicted, its superpage is demoted. The system prevents write to clean superpages. If a process tries to write to a clean SP, demote it and re-promotes it later only if all base pages are used. As per the experimental results,the implemented superpage management system achieves 30%-60% improvement in performance. Thus the authors argue that this approach is a transparent and effective solution to the problem of superpage management in operating systems which can be integrated into an existing OS .