

Sharing Data in Page-Based Intelligent Memory

Mark Oskin, Timothy Sherwood, Justin Hensley, Sinclair Yeh, and Frederic T. Chong
Department of Computer Science
University of California at Davis

Processors and memory suffer from a growing gap in performance. To address this gap, many researchers have proposed intelligent memory systems. These proposals, however, largely neglect communication issues between memory chips and between memory and disk. We propose Active Pages, a page-based model of intelligent memory that replaces conventional memory but leaves in place the existing mechanisms that support virtual addressing and multiple processors. Furthermore, Active Pages support multiple memory chip designs. An Active Page consists of a super-page of data and a set of associated functions that operate on that data. For example, pages may store an array and functions may include insert, delete, and find. Active Pages allocated by the same user process each form part of a global virtual address space. Functions for each page may share data with other pages by accessing the address on those pages.

We introduce RADram, an implementation of Active Pages which integrates reconfigurable logic and DRAM technology. Each Active Page is implemented by a block of reconfigurable logic (approximately 256 4-Luts) placed next to the sense amps on a DRAM sub-array (each 512 Kbytes in a 1 Gigabit DRAM). Active-Page functions are implemented by the reconfigurable logic and use virtual addresses. Each page contains a base register to all detection and translation of virtual address references between the function and their local pages. References that fall outside of the local page are queued and a processor interrupt is triggered. The processor then satisfies requests queued from all active pages by simply reading and writing between pages. If a page is swapped out to disk, a page fault is triggered. This processor-initialized approach to sharing data is expensive, but it greatly simplifies design of the architecture and operating system.

Simulation results show up to a 1000X performance improvement on a dedicated Active-Page system as compared to a conventional memory system. Multi-programmed workloads running on our prototype operating system, ActiveOS, also show substantial speedups even as pages are swapped to and from disk. This talk will focus upon scheduling and page replacement algorithms for efficient sharing of data between Active Pages. We will also discuss hardware alternatives to processor-initiated communication.