Interference Aware Cache Designs for Operating System Execution

David Nellans, Rajeev Balasubramonian, Erik Brunvand

UUCS-09-002

School of Computing
University of Utah
Salt Lake City, UT 84112 USA

February 3, 2009

Abstract

Large-scale chip multiprocessors will likely be heterogeneous. It has been suggested by several groups that it may be worthwhile to implement some cores that are specially tuned to execute common code patterns. One such common application that will execute on all future processors is of course the operating system. Many future workloads will spend a large fraction of their execution time within privileged mode, either executing system calls or pure operating system functionality. Vast transistor budgets and relatively low on-chip communication latencies make it feasible to off-load the execution of privileged instruction sequences on to such a custom core. In this paper, we first examine this off-load approach and attempt to understand its benefits. We then try to architect a solution that captures the benefits of off-loading and eliminates its disadvantages. In essence, the benefits of off-loading can be attributed to reduced cache interference, while its disadvantages are the high latency costs for off-load and cache coherence. Our proposed solution employs a special OS cache per core and improves performance by up to 18% for OS-intensive workloads without any significant addition of transistors. We consider several design choices for this OS cache and argue that it is a better use of transistor and power budget than the off-loading approach when both adding to the transistor budget or leaving it unchanged.