



*Improving Application Timing Predictability and Caching Performance on Multi-core Systems*

J. Xiao

Modern computing systems are constructed using commodity multi-core processors, on which part of the memory subsystem is shared by different cores on the same processor. Multiple applications executing simultaneously on a multi-core system contend for the shared memory resources, such as last level caches (LLC) and main memory, causing inter-application interference. Such inter-application interference, if uncontrolled, results in unpredictable execution delay for individual applications and severe system performance degradation. In this dissertation, we focus on shared cache interference and investigate two issues raised by the increasing complexity of underlying hardware and software for multi-core systems: timing predictability of real-time computing and caching performance for high performance computing.

In the first research line, we improve the timing predictability for embedded multi-core system by simulation and analytical approaches for the timing analysis. With regard to the simulation approach, we developed SysRT, a simulator of real-time operating systems (RTOS) that allows developers and researchers to easily explore and validate embedded RTOS design alternatives. The simulator contains different types of application models and a modular RTOS kernel model, all developed in SystemC. Efficient and precise modeling of preemptive scheduling is achieved via an event-driven simulation approach, allowing simulations to be performed much faster than cycle-accurate simulations. SysRT outperforms state-of-art simulators in both simulation speeds and accuracy. We also demonstrate the flexibility of SysRT and its benefits in evaluation of timing performance for software tasks .

With regards to the analytical approach, we first develop a new schedulability analysis of global scheduling for real-time multi-core systems with shared caches. We construct an integer programming formulation and an iterative algorithm to obtain the upper bound on shared cache interference a task may exhibit during one job execution. The upper bound on shared cache interference is subsequently integrated into the schedulability analysis to derive a new schedulability condition for global scheduling. Later, we extend the schedulability analysis for the partitioned scheduling. We propose a novel cache-interference aware task partitioning algorithm, called CA-TPAR. We conduct schedulability analysis of CA-TPAR and formally prove its correctness. A range of experiments is performed to investigate how the schedulability is degraded by shared cache interference. We also evaluate the schedulability performance of global scheduling (Earliest Deadline First and Fixed Priority) against CA-TPAR over randomly generated tasksets.

In the second research line, We propose CP\$\_{pf}\$, a prefetch aware LLC partitioning approach to improving LLC management for high performance caching. We first study the interaction between hardware prefetching and LLC management by analyzing the variation of application performance when varying the effective LLC space in the presence and absence of hardware prefetching. Motivated by this study, we then classify applications into two categories, prefetching sensitive (PS) and non prefetching sensitive (NPS) applications, by the performance benefit they experience from hardware prefetchers. CP\$\_{pf}\$ consists of a method using Precise Event-Based Sampling techniques for the online classification of PS and NPS applications and a cache partitioning scheme using Cache Allocation technology to distribute the cache space among PS and NPS applications. The prototype of CP\$\_{pf}\$ is implemented as a user-level runtime system on Linux. Finally, we show the system performance improvement achieved by CP\$\_{pf}\$.