CCIT REPORT #701 August 2008

Multiple Multithreaded Applications on Asymmetric and Symmetric Chip MultiProcessors

Tomer Y. Morad† Avinoam Kolodny† Uri C. Weiser†

†Department of Electrical Engineering
Technion, Haifa, Israel
{tomerm@tx, kolodny@ee, uri.weiser@ee}.technion.ac.il

Abstract

This paper evaluates new techniques to improve performance and efficiency of Chip MultiProcessors (CMP) for workloads consisting of multiple multithreaded applications. Multithreaded applications contain serial phases (single thread) and parallel phases (many threads). While scheduling threads, current techniques do not differentiate between these two phases, resulting in sub-optimal usage of the multiprocessor resources. In this paper, we propose a new thread scheduling mechanism which takes into account the different requirements of each phase, granting higher priority to applications during their critical-serial phases. The advantages of the proposed scheduling mechanism, shown by analytical and experimental evaluation, are threefold. First, system throughput and power efficiency are improved by making better use of the available multiprocessor computing power. Some of the benchmarks show system throughput improvements of as much as 16%. Second, fairness in resource allocation between the applications is improved by as much as 26%. Third, the jitter in execution runtimes in different runs of the same set of applications is reduced by up to 88%. The analysis is performed for asymmetric multiprocessors, where some of the computing cores are faster than others, as well as for symmetric multiprocessors in which all cores are identical. All experiments in this paper are performed in a real environment, consisting of full benchmarks running on a real multiprocessor and operating system.

1 Introduction

Multithreaded applications can take advantage of the added computing ability offered by today's multiprocessors by executing in parallel on many cores. With an ever-increasing core population embedded in state-of-the-art systems [15], the use of multithreading in applications is expected to increase. In this paper, we strive to improve system performance as measured by several metrics when scheduling multiple multithreaded applications in parallel on asymmetric multiprocessors (where some computing cores are faster than others), as well as on symmetric multiprocessors (where all cores are identical).

When examining multithreaded applications, one can identify two types of execution phases, serial phases and parallel phases. In serial phases only one thread is active, whereas parallel phases are comprised of many concurrently active threads. Typically, data preparation for the parallel phases and inherently sequential calculations are done in the serial phases. The heavy independent calculations are performed in the parallel phases.

When two multithreaded applications are run simultaneously, the serial thread of one application may be available for execution together with the parallel threads of the other application. Fig. 1 shows an example of the four possible joint states of two multithreaded applications. The vertical axis represents time, advancing from top to bottom. At each point in time, the number of active threads for each application is shown.

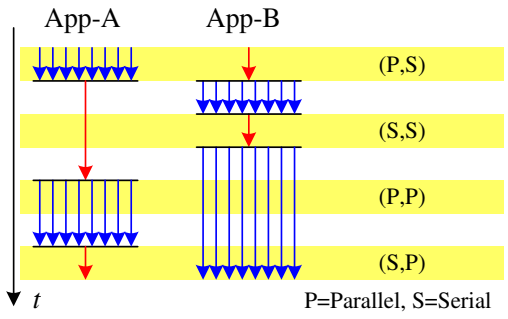

Fig. 1. Illustration of joint states of two sample applications running simultaneously.