Chapter 14
A Machine Learning Based Meta–Scheduler for Multi–Core Processors

Jitendra Kumar Rai
University of Hyderabad and ANURAG, India

Atul Negi
University of Hyderabad, India

Rajeev Wankar
University of Hyderabad, India

K. D. Nayak
ANURAG, India

ABSTRACT
Sharing resources such as caches and memory buses between the cores of multi-core processors may cause performance bottlenecks for running programs. In this paper, the authors describe a meta-scheduler, which adapts the process scheduling decisions for reducing the contention for shared L2 caches on multi-core processors. The meta-scheduler takes into account the multi-core topology as well as the L2 cache related characteristics of the processes. Using the model generated by the process of machine learning, it predicts the L2 cache behavior, i.e., solo-run-L2-cache-stress, of the programs. It runs in user mode and guides the underlying operating system process scheduler in intelligent scheduling of processes to reduce the contention of shared L2 caches. In these experiments, the authors observed up to 12 percent speedup in individual as well as overall performance, while using meta-scheduler as compared to default process scheduler (Completely Fair Scheduler) of Linux kernel.

INTRODUCTION
Multi-core processors have multiple execution cores in each processor package. Topology of multi-core processors involves sharing of various resources like caches and memory bus between the cores. The sharing of these resources brings new opportunities and challenges for system software. For example programs running on cores which share L2 cache, may compete for space in the L2 cache, whereby may cause eviction of cache lines allocated to each other. This causes degradation
in performance of the applications running on multi-core processors due to poor performance isolation (Fedorova et al., 2005; Fedorova et al., 2007; Zhao et al., 2007).

Different programs have different requirements of space in L2 cache, which itself change along with the progress of programs during runtime. Hence achieving optimal performance requires optimizations to be done in a dynamic and adaptive manner.

Process scheduling plays an important role in allocation of cpu resources to running programs. An ideal scheduler should consider the requirement as well as contention for the resources for making the scheduling decisions. Thus system software components like process scheduler should be made aware of multi-core topologies and the characteristics of the processes (Siddha et al., 2007) to get next level of performance.

In this paper we describe a meta-scheduler for reducing the contention for shared L2 caches on multi-core processors. The meta-scheduler takes into account the multi-core topology as well as the L2 cache related characteristics of the processes. It uses the model generated by the process of machine learning to predict the L2 cache behavior (solo-run-L2-cache-stress) of the programs. The meta-scheduler runs in user mode and guides the underlying operating system process scheduler in intelligent scheduling of processes to reduce the contention of shared L2 caches.

METRIC TO CHARACTERIZE L2 CACHE BEHAVIOUR

The meta-scheduler uses the solo-run-L2-cache-stress of programs to characterize their L2 cache behaviors. Solo-run-L2-cache-stress of a program is the number of misses and pre-fetches of L2 cache per instruction, experienced by that program while not sharing the L2 cache with programs running on other core.

The L2 cache stress observed (using the hardware performance counters) for a program running on a core is different from its solo-run-L2-cache-stress, if there is another program running on other core sharing the L2 cache with first. This is so because the observed L2 cache stress results from the interaction of their individual cache access patterns and working sets based on their locality. The meta-scheduler uses regression model to deduce the solo-run-L2 cache stress of running programs. The regression model was built off-line through the process of training machine learning algorithm.
Related Content

Management Scheme for Data Collection within Wireless Sensor Networks
[www.igi-global.com/article/management-scheme-data-collection-within/66049?camid=4v1a](www.igi-global.com/article/management-scheme-data-collection-within/66049?camid=4v1a)

Alignment: The Activity Domain in the Centre
[www.igi-global.com/chapter/alignment-activity-domain-centre/39681?camid=4v1a](www.igi-global.com/chapter/alignment-activity-domain-centre/39681?camid=4v1a)

In Search of an Integrating Construct
[www.igi-global.com/chapter/search-integrating-construct/39682?camid=4v1a](www.igi-global.com/chapter/search-integrating-construct/39682?camid=4v1a)

Applying Probabilistic Adaptation to Improve the Efficiency of Intra-Query Load Balancing
[www.igi-global.com/article/applying-probabilistic-adaptation-improve-efficiency/75548?camid=4v1a](www.igi-global.com/article/applying-probabilistic-adaptation-improve-efficiency/75548?camid=4v1a)