

## Chapter 3.11

# 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.

DOI: 10.4018/978-1-60960-818-7.ch3.11

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.

## **OVERVIEW OF THE META-SCHEDULER**

The meta-scheduler relies on reducing the contention between the programs for level-2 (L2) caches, while those programs run on the cores which share the L2 caches. In turn it also reduces the

contention for front side bus shared by cores as on Intel multi-core processors. The meta-scheduler utilizes the hardware performance counters and offline built regression model to know about the L2 cache behavior of running programs. Hardware performance counters are special purpose registers, which are provided by the Performance Monitoring Unit (PMU) of modern processors. The hardware performance counters can be configured to measure various performance events of interest like number of instructions retired, L2 cache misses etc. Details regarding the number of performance counters and the way to configure them are described in processor manuals (Intel Manuals).

The meta-scheduler periodically samples the data from the hardware performance counters for the programs running on a multi-core processor; and guides the scheduling decisions of underlying operating system process scheduler to alleviate the contention for shared last level cache (L2-cache). The scheduling decisions are made in dynamic manner as mentioned under section mentioning policy framework of meta-scheduler.

## **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

11 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:  
[www.igi-global.com/chapter/machine-learning-based-meta-scheduler/56161](http://www.igi-global.com/chapter/machine-learning-based-meta-scheduler/56161)

## Related Content

---

### Formal Rules for Fuzzy Causal Analyses and Fuzzy Inferences

Yingxu Wang (2012). *International Journal of Software Science and Computational Intelligence* (pp. 70-86).  
[www.irma-international.org/article/formal-rules-for-fuzzy-causal-analyses-and-fuzzy-inferences/88928](http://www.irma-international.org/article/formal-rules-for-fuzzy-causal-analyses-and-fuzzy-inferences/88928)

### Designing a Hybrid Approach for Web Recommendation Using Annotation

Sunny Sharma, Vijay Rana and Vivek Kumar (2022). *Applications of Computational Science in Artificial Intelligence* (pp. 234-247).  
[www.irma-international.org/chapter/designing-a-hybrid-approach-for-web-recommendation-using-annotation/302069](http://www.irma-international.org/chapter/designing-a-hybrid-approach-for-web-recommendation-using-annotation/302069)

### Machine-Learning-Based White-Hat Worm Launcher in Botnet Defense System

Xiangnan Pan, Shingo Yamaguchi, Taku Kageyama and Mohd Hafizuddin Bin Kamilin (2022). *International Journal of Software Science and Computational Intelligence* (pp. 1-14).  
[www.irma-international.org/article/machine-learning-based-white-hat-worm-launcher-in-botnet-defense-system/291713](http://www.irma-international.org/article/machine-learning-based-white-hat-worm-launcher-in-botnet-defense-system/291713)

### Graph and Neural Network-Based Intelligent Conversation System

Anuja Arora, Aman Srivastava and Shivam Bansal (2019). *Nature-Inspired Algorithms for Big Data Frameworks* (pp. 339-357).  
[www.irma-international.org/chapter/graph-and-neural-network-based-intelligent-conversation-system/213043](http://www.irma-international.org/chapter/graph-and-neural-network-based-intelligent-conversation-system/213043)

### Optimization of a Solar-powered Irrigation System

(2020). *Multi-Objective Optimization of Industrial Power Generation Systems: Emerging Research and Opportunities* (pp. 63-109).  
[www.irma-international.org/chapter/optimization-of-a-solar-powered-irrigation-system/246402](http://www.irma-international.org/chapter/optimization-of-a-solar-powered-irrigation-system/246402)