Chapter 17
Using Machine Learning Techniques for Performance Prediction on Multi-Cores

Jitendra Kumar Rai
ANURAG, Hyderabad, India

Atul Negi
University of Hyderabad, India

Rajeev Wankar
University of Hyderabad, India

ABSTRACT

Sharing of resources by the cores of multi-core processors brings performance issues for the system. Majority of the shared resources belong to memory hierarchy sub-system of the processors such as last level caches, prefetchers and memory buses. Programs co-running on the cores of a multi-core processor may interfere with each other due to usage of such shared resources. Such interference causes co-running programs to suffer with performance degradation. Previous research works include efforts to characterize and classify the memory behaviors of programs to predict the performance. Such knowledge could be useful to create workloads to perform performance studies on multi-core processors. It could also be utilized to form policies at system level to mitigate the interference between co-running programs due to use of shared resources. In this work, machine learning techniques are used to predict the performance on multi-core processors. The main contribution of the study is enumeration of solo-run program attributes, which can be used to predict concurrent-run performance despite change in the number of co-running programs sharing the resources. The concurrent-run involves the interference between co-running programs due to use of shared resources.

DOI: 10.4018/978-1-4666-2065-0.ch017

INTRODUCTION

Multi-core has become the dominant processor architecture at present. Majority of the computing systems are based on multi-core processors, which include both the desktops as well as servers. In future the number of cores on a single processor chip is going to increase with the upcoming generations of processors. Most of the recent multi-core processors have last level caches, which are shared among the cores (Intel Corporation; Kongetira, Aingaran, & Olukotun, 2005; Golla, 2006). Programs co-running on the cores of multi-core processors also share other resources such as hardware prefetch unit, Front Side Bus (FSB) and memory controller along with last level caches. Along with the growth in number of cores per chip with the generations of processors, future multi-core based systems are poised to witness increased degree of sharing of the resources.

The interference between programs co-running on the cores sharing the resources causes degradation in the performance of systems based on multi-core processors. For example a process on one core may cause the eviction of data belonging to process on the other core, with which it shares the cache space. Such interference between co-running programs due to use of shared cache space can cause the performance of simultaneous running processes to get affected by each other. We measured the performance of some of programs from SPEC cpu2006 benchmark suite (SPEC, 2006) on our Intel quad-core Xeon X5482 processor based experimental platform (described in section named EXPERIMENTAL PLATFORMS) for their solo-run as well as concurrent-run. The values of solo-run and concurrent-run performance for those programs in terms of cycles per instruction (CPI) are mentioned in Table 1.

The performance results shown in Table 1 indicate towards degradation in performance of the programs in concurrent-run as compared to their solo-run (i.e., increase in concurrent-run CPI as compared to solo-run CPI). It can be seen that concurrent-run performance of the same program differs when it is run along-with different co-runner programs. In Table 2 we mention the co-runner based performance degradation of the programs mentioned in Table 1. The co-runner based performance degradation results are also shown in Figure 1.

The degradation in performance results from the interactions of program and its co-runner behaviors with respect to usage of resources shared by the cores. In our previous work (Rai et al., 2010) we used the solo-run attributes of the programs to predict their concurrent-run performance on multi-core processors. The methodology proposed in the work to build the model utilized machine learning techniques. The trained model predicts concurrent-run performance of a program

Table 1. Solo-run and concurrent-run performance of some of the SPEC cpu2006 programs on Intel Xeon X5482 processor

Program Names	Performance in terms of CPI, for			
	Solo-run	Concurrent-run, when co-runner is		
		429.mcf	433.milc	459.GemsFDTD
429.mcf	5.89	8.37	10.15	10.04
433.milc	2.41	2.89	3.46	3.54
459.GemsFDTD	1.62	1.98	2.21	2.39
462.libquantum	1.67	2.24	3.05	3.06
410.bwaves	1.17	1.36	1.45	1.53