Thermal Prediction and Scheduling of Network Applications on Multicore Processors

Chih-Hsun Chou
UC Riverside
College of Engineering
Department of Electrical Engineering
chchou@engr.ucr.edu

Mehmet E. Belviranli
UC Riverside
College of Engineering
Department of Computer Science and Engineering
belviram@cs.ucr.edu

Laxmi N. Bhuyan
UC Riverside
College of Engineering
Department of Computer Science and Engineering
bhuyan@cs.ucr.edu

ABSTRACT

As processor power density increases, chip/core temperature control becomes critical for building multicore systems. This paper addresses the problem of inter-core thermal coupling and periodic thermal variation while executing multi-threaded network applications in a multicore architecture.

1. INTRODUCTION

The network industry is aggressively scaling the number of processors to meet the challenge of high bandwidth low latency packet processing. However, power and thermal constraints pose a significant challenge to network system design. Network packet processing is a periodic task execution, where the core temperature rises as the packet is processed and falls as the packet leaves. An earlier paper [1] modeled this rise and fall in a multicore processor assuming that there is no heat flow between the cores. In this paper, we extend the model to realistic multicore architectures assuming such heat flow between the cores. Then we develop thermal aware scheduling for network applications to balance the temperature and thus reduce the power consumption.

2. Preliminary

2.1 System Architecture

Figure 1 shows an overview of the targeted system architecture, consisting of multiple network processors (NPs) with local queues of tasks. The scenario represents the arrival of a periodic task, where the execution takes place when a packet arrives. The scheduler consists of three functional modules, namely packet dispatcher, thread monitor and thermal manager. In each scheduling cycle, the packet dispatcher fetches the next available packet from the global queue, makes the scheduling decision, and then dispatches the packet into the local queue of the appropriate core. The thread monitor keeps track of the thread activity of every network processor. In addition, the thread monitor reads the temperature reading from the on chip temperature sensors. The above information will be used by the packet dispatcher and thermal manager to make proper scheduling decision. The thermal manager makes decision for applying the thermal management techniques, e.g. Stop&GO and Thread migration, when the temperature reaches the trigger temperature, and is based on our predictive thermal model and the core/cache activity recorded by the thread monitor.

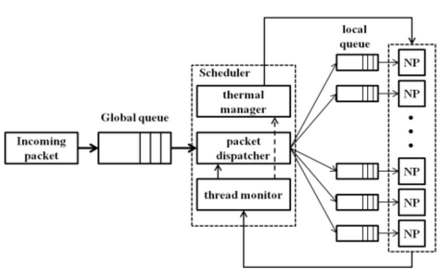

Figure 1. Overview of the system architecture.

2.2 Thermal Model for Periodic Task on Multi-core Architecture

In the multicore architecture, the thermal variation is not only caused by heating and cooling of the core, but also affected by the thermal behavior of the neighboring core. Figure 2 shows our proposed multi-core thermal model based on the general mesh core topology. Each pair of parallel RC represents the thermal model for the core [2]. We let the thermal resistance R_{ij} be the lateral heat resistance between $core_i$ and $core_j$, which represents the lateral heat transfer.

The network packet processing is a continuous flow consisting of interleaved temperature rising and falling phases, and can be seen as periodic tasks. From our proposed thermal model, the temperature can be derived from the most recent temperature history, the timing, and the power consumption of packet processing. By recursively model the temperature rising and falling, we can model the periodic task execution. We proposed three different modeling techniques for the periodic task execution: 1) upper bound, 2) lower bound and 3) average approximation. For the upper bound approximation, we assume the neighboring core is

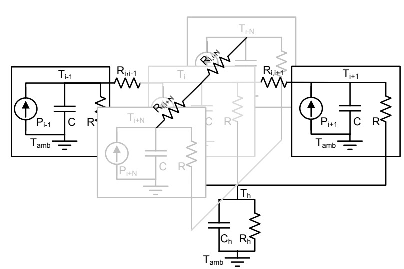

Figure 2. RC thermal model for multi-core.

978-1-4799-1640-5/13/$31.00 ©2013 IEEE

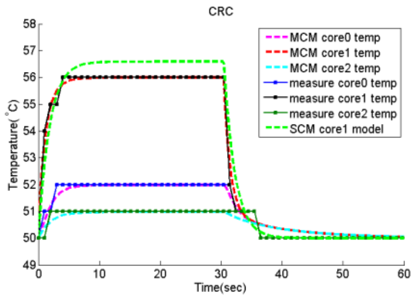

Figure 3. Lateral heat transfer model for CRC.

always busy. The lower bound approximation assumes the neighboring core is always idle and the average approximation assumes all its neighbors are running with the average power over its execution.

3. Preliminary Result

3.1 Model Verification

To have a thermal model for our experiment platform, Intel server with two Quad-Core Xeon E5335 processors, we let the lateral heat resistance in the Y-direction, as shown in Figure 2, to be infinite. In order to see the importance of topology-awareness, which considers the lateral heat transfer, we compare the single core model (SCM), which does not consider neighbors, with our multi-core model (MCM). The RC parameter of SCM is derived from the result shown in [1]. The temperature profile of sensor reading and model prediction for *CRC* is shown in Figure 3. The temperature of the neighboring cores (C0,C2) are affected due to heat transfer. We can see that the single core thermal model fails to predict the lateral heat transfer to neighboring cores. It predicts higher temperature because it ignores the inter-core thermal transfer, that reduces the temperature. The effect will be just opposite if core0 executes another thread along with core0. Our topology-aware thermal model adequately represents the thermal characteristics for single task in both phases. Although the temperature sensor only returns discrete integer values, we still observe that the average difference is only 0.2°C, and the maximal difference is just 0.4°C.

Figure 4 shows the three different approximation model for periodic task execution on *URL* of core 1. The sawtooth temperature trace is caused by the periodic task execution. The actual temperature falls in the range of upper and lower bound approximation. When the traffic is high, which means the inter-arrival time is short, the upper bound approximation will have higher accuracy. In the contrary, the temperature reading is closer to the lower bound when the traffic is low. Also, the average approximation will have good accuracy when the traffic is steady.

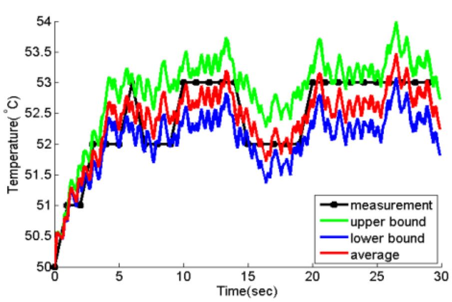

Figure 4. Periodic task thermal model for URL.

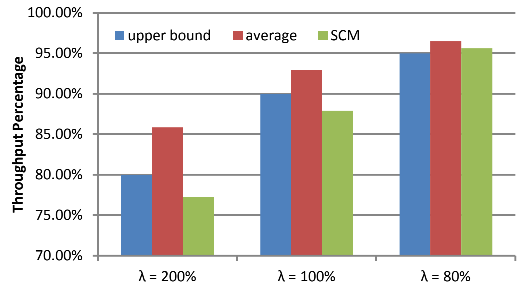

Figure 5. Throughput under different incoming traffic.

3.2 Performance Evaluation

We consider the throughput as our performance requirement. All our scheduler schemes suffer from throughput degradation to some extent due to enforced thermal constraint. We conduct extensive experiment with NetBench benchmark suite [3] on the throughput percentage under three different incoming traffic rates. We insert exponential distribution packet inter-arrival time between packets with three different mean arrival rate (λ), 200%, 100% and 80% of the total service rate. Besides, the thermal constraint is set to be 55°C, and the trigger temperature is 0.2°C. The results of the throughput percentage for different incoming traffic are shown in Figure 5. We can see that the throughput percentage increases, from average 86.3% to 96.2% when the incoming traffic rate decreases, from 200% to 80% of total service rate. The reason is because that when the incoming traffic rate is lower, the inter-arrival time between packets is longer. So the overhead caused by the Stop&Go is smaller. Another observation is that the average approximation out performs the other two. The reason is that the upper bound approximation overestimate the temperature, hence introduce longer Stop&Go suspension time then needed. For the SCM, since it may underestimate the temperature, we need to set a lower trigger temperature, 0.5 °C below the temperature constraint in this case, so that the thermal constraint is always satisfied. As a result, the throughput performance will be compromised.

4. Acknowledgement

The authors gratefully acknowledge the support of this work by National Science Foundation under grant CNS-0832108 and CNS-1216014.

5. References

[1] J. Kuang, L. Bhuyan "Thermal-aware Scheduling of Network Applications on Multicore Architecture," *Proceedings of the 14th ACM/IEEE ANCS*, pp. 1-14, 2009.

[2] K. Skadron, T. Abdelzaher, M.R. Stan, Control-theoretic techniques and thermal- RC modeling for accurate and localized dynamic thermal management, in Proc. of HPCA, 2002.

[3] G. Memik, W. H. Mangione-Smith, and W. Hu, "Netbench: A benchmarking suite for network processors," in Proc. Of ICCAD'01, 2001.