Journal of Systems Architecture 62 (2016) 54–62

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

A thermal-aware scheduling for multicore architectures

Ting-Hsuan Chien, Rong-Guey Chang*

Department of Computer Science and Information Engineering and the Advanced Institute for Manufacturing with High-tech Innovations, National Chung Cheng University, Chia-Yi, Taiwan

ARTICLE INFO

Article history:
Received 9 March 2015
Revised 24 September 2015
Accepted 14 December 2015
Available online 22 December 2015

Keywords:
Dynamic thermal management
Multicore
Thermal-aware scheduling

ABSTRACT

With the advance of technology, the power density (temperature) increases rapidly to threaten system performance, reliability, and even system safety. Development of a thermal management method to reduce thermal hotspots and distribute the temperature uniformly has become an important issue. Therefore, dynamic thermal management (DTM) has emerged as an effective technique to remedy these issues above. In this paper, we propose a proactive thermal management scheme on the Criticore platform developed by our team to avoid suffering high temperature of the system. The proposed approach can schedule threads to prevent the system from overheating with the aid of the thermal sensors and the Power Management Circuit (PMC) designed in the Criticore. Furthermore, a novel thread migration is also presented to increase the reliability of the system.

1. Introduction

With the advance of technology, the power density (temperature) increases rapidly to threaten system performance, reliability, and even system safety in many ways [1,10]. First, for high temperature, the cooling cost increases quickly. Second, power leakage is exponentially related to temperature; that is, it increases as the temperature rises. Next, high temperature degrades the performance of a system. Finally, thermal hotspots and steep temperature gradients result in an adverse effect on reliability, thus shortening the lifecycle severely. As a result, development of a thermal management method to reduce thermal hotspots and to distribute the temperature uniformly has become a crucial issue.

Several dynamic thermal management (DTM) techniques have been proposed to cool thermal hotspots on the chip, like fetch throttling [1,8] and dynamic voltage and frequency scaling (DVFS) [1,13]. They are reactive in nature to take action after the temperature reaches a certain level. Although they can cool down thermal hotspots on chip, they result in significant performance degradation. They do not address temperature reduction as much as possible to minimize the impact on reliability. Besides, conventional DTM techniques do not focus on balancing the temperature spatially or temporally. They may produce thermal gradients or cycles created by workload rate and power management decisions. Worse, for systems with dynamic power management (DPM) that turn off cores, it will accelerate the reliability degradation of a system [12].

However, proactive DTMs may be effective in temperature control against thermal emergency, since they trigger the control schemes to prevent the system from reaching the emergent temperature threshold. For example, HybDTM [7] dynamically adjusted process priorities based on estimated individual temperatures. Timeslice was calculated based on priority, which was a measure of the maximum CPU time. They assigned shorter timeslices to “hot” processes which can increase the core temperature and longer ones to “cold” processes which can decrease the core temperature. Nevertheless, this will lead to the hot tasks being pushed to the last execution. Several proactive DTMs based on runtime task migration have been proposed [5,16]. Yeo et al. [16] proposed an approach that did not try to maintain a balanced workload among processors. The approach worked efficiently only when the number of tasks was less than the number of cores. Ge et al. [5] proposed an approach that achieved balancing thermal profile by proactive task migration among neighboring cores. It might be advisable to perform the migration frequently at the cost of performance degradation.

In this paper, the threads are profiled to be divided into hot threads, warm threads, and cold threads, We then propose a predictive thermal-aware scheduling to let hot threads to execute as long as possible in advance. To avoid producing high temperature, the reactive approach is applied to reduce the temperature of a hot core. The proposed temperature-aware scheduler is scalable to any

* Corresponding author. Tel.: +886 5 272 0411.
E-mail addresses: cth103p@cs.ccu.edu.tw (T.-H. Chien), rgchang@cs.ccu.edu.tw (R.-G. Chang).

http://dx.doi.org/10.1016/j.sysarc.2015.12.003
1383-7621/

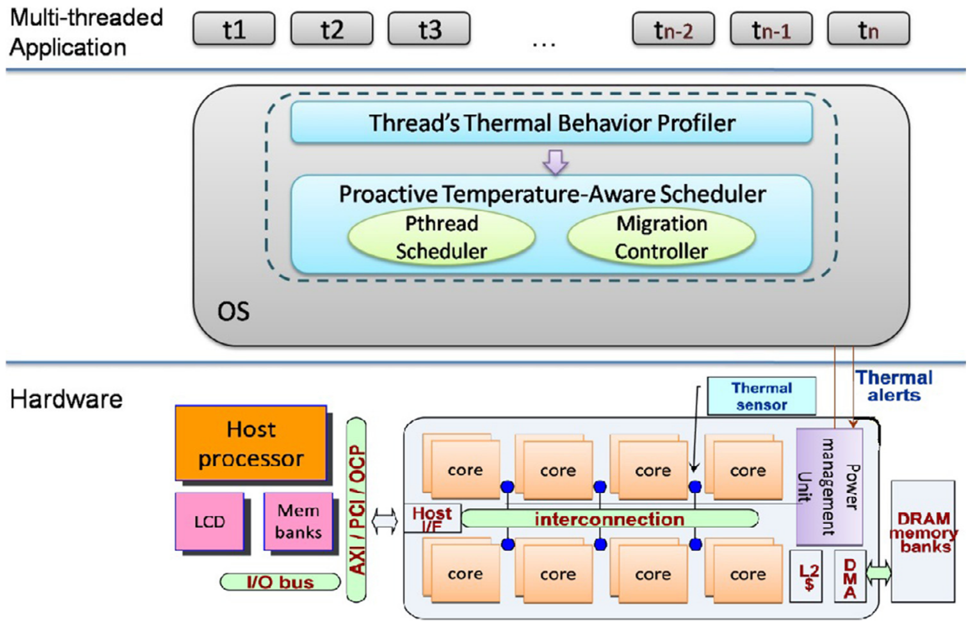

Fig. 1. System overview.

number of cores. The main idea of this paper is summarized as follows:

- Threads are classified into three categories based on temperature characteristics and temperature variation of a core while executing a thread. The categories of threads will be adjusted dynamically at run time to make the predictions more accurate.
- Fig. 1 is the overview of our system architecture. For each core, we have designed and implemented thermal sensors to obtain its temperature. The temperature information will be sent to Power Management Circuit (PMC) to issue an alarm based on θ_{dan}, θ_{cl} and θ_{ch}, which is defined later in this paper. The design of thermal sensors and PMC is not the scope of this paper. The result shows that our scheme can reduce the temperature of our system and achieve thermal balancing among cores.
- The thread migration mechanism is conducted to ensure the reliability of the system when a core reaches a critical thermal threshold.
- The proposed approach can be adaptive while available core changes during execution.

The rest of this paper is organized as follows. Section 2 presents the related work and Section 3 presents the propose scheduling approach. Section 4 shows the implementation detail and scenarios at run time. The experimental results are shown in Section 5 and Section 6 concludes this paper.

2. Related work

Many techniques are proposed to manage the temperature of a chip and to mitigate the temperature rising, such as dynamic thermal management.

2.1. Load balance for dynamic scheduling algorithm

DTM is a common technique to control the temperature of a chip in a safe temperature region. In the temperature graph, there are two important horizontal dotted lines: one is the threshold temperature, and the other is the trigger temperature. The trigger temperature represents when the DTM technique is triggered once the temperature of a chip arrives at the trigger temperature. If the temperature of a chip is over the threshold temperature, it represents that system is in a critical situation.

HybDTM [7] can be used reactively to deal with thermal emergencies, low overhead software techniques can be used to lower system temperature that relies on application specific thermal profiles. They dynamically adjust process priorities based on their estimated individual temperatures. And they calculate timeslice based on priority, which is a measure of the maximum CPU time that a process is allowed to run in an epoch. They assign shorter timeslices to “hot” processes and longer ones to “cold” processes.

Paci et al. [11] proposed a model to investigate the need for temperature aware design in a low-power systems-on-a-chip and provide guidelines to delimit the conditions for which temperature-aware design is needed. By using the model to compute the heat conductance and capacitances, the hot spot can be avoided.

Ge et al. [5] proposed a framework for distributed thermal management for many core systems where it achieved the balancing thermal profile by proactive task migration among neighboring cores. They divided the task into two modes according to their temperature prediction, master and slave. A master DTB agent issues a task migration request to its nearest neighbors which are DTB slaves. The goal is to ensure that each processor has a good mix of hot tasks and cold tasks.

2.2. Thermal-aware scheduling

The Software methods can play a critical role in the thermal management problem. Software thermal management usually applied various scheduling policies to reduce temperature.

Xie and Hung [15] proposed both power-aware and thermal-aware approaches to the task allocation and scheduling and showed that the thermal-aware approach outperforms the power-aware schemes in temperature reductions.

Choi et al. [3] considered the trade-off between temporal and spatial hotspots mitigation schemes and thermal time constants, their schemes enabled lowering of on-chip unit temperature by changing the workload in a timely manner with the operating system and existing hardware support. Their scheduler used three techniques that have different triggering conditions and response