Dynamic Power Management and Dynamic Voltage Scaling in Real-time CMP Systems

Lei Miao, Yong Qi, Di Hou, Chang-li Wu, Yue-hua Dai
1. School of Electronics and Information Engineering, Xi’an Jiaotong University
2. Chinese Society of Astronautics
leimiao@163.com

Abstract

Real-time embedded systems need reducing power consumption while still maintaining high performance. Dynamic power management (DPM) and dynamic voltage scaling (DVS) are effective control techniques to reduce the energy consumption. We present a real-time energy saving scheduling (RESS) algorithm using DPM and DVS to reduce the energy cost for chip multiprocessor (CMP) systems. The simulating experiment results indicate that our algorithm can save system energy cost with little system performance degradation.

1. Introduction

Recently chip multiprocessors are becoming increasingly popular in embedded domain. So the energy saving of CMP system is becoming an interesting issue for many researchers. Dynamic power management (DPM) and dynamic voltage scaling (DVS) have been employed as available techniques to reduce the energy consumption of CMOS microprocessor system. DPM changes the power state of cores on chip to lower the energy consumption according to the performance constraints [1]. DVS has been emerged as an effective policy to save energy by varying the operating voltage and frequency of cores during run time [2].

In recent years, there were studies on DVS & DPM-based scheduling for CMP system energy reduction. Most of the methods are not designed for real-time system and need information such as release time, execution time and deadline in advance. We propose a two-stage optimal energy saving scheduling algorithm to reduce the system energy consumption in real-time. We use real-time energy saving scheduling algorithm (RESS) to reduce the system energy cost with little system performance penalty.

2. System model

The system model consists of the job arrived, power manager (PM), frequency controller (FC) and chip multiprocessor system.

The hardware architecture model we used includes eight Alpha 21264 (EV6) processors simulated by Simplescalar 3.0 [3]. Each processor core has private L1 instruction and data caches. All cores share an 8MB on-chip L2 cache through a common bus. Each core has three states: active, idle, sleep. Core enters the idle state as soon as it finished the last job.

The relationships between operating frequency, system energy cost, and supply voltage are shown as following [4]

\[ f \propto \xi \cdot V_{dd} \]  
\[ E \propto \alpha \cdot V_{dd}^2 \]

We can see that the reduction in frequency combined with a linear reduction of operating voltage results in an approximately quadratic reduction of the energy consumption. We can achieve the quadratic system energy saving at the expense of prolonging the execution time linearly.

We assume a dynamic priority multiprocessor system which has an independent and preemptive task set. The job arrival behavior follows Poisson process. The probability of k jobs arrived during time interval t can be described as

\[ P(t, k) = e^{-\lambda t} \frac{(\lambda t)^k}{k!} \]  

Where, \( \lambda \) is an average job arrival rate.

3. Real-time energy saving scheduling algorithm

We propose an energy saving scheduling algorithm which includes two energy saving stages. The PM
modulates the state of cores in the system level. Frequency controller scales the operating frequency during the runtime of the core.

Power management decides the number of active core according to the tradeoff between the system performance and the power consumption. Every active core maintains a job queue. Power manager allocates arrival jobs to the active cores according to their queue length. PM monitors the queue lengths of each core and balances the load between cores periodically. PM adjusts the state of cores based on the arrival job and the queue length of cores. When a job arrived PM decides which core the job allocates to. If all the active cores are full occupied it will wake up a sleep core to deal with the job. If PM observes there are idle cores and cores with very long queue at same time, it will migrate jobs from the busy core to the idle one. When the last active core finished its jobs PM decides whether to leave it in idle state or places it into sleep state. If PM kept it on idle state, the core will wait until the first job arrived. When a job arrived, it transfers into active state immediately. If the PM decided to switch it into sleep state, the system transfers the core from idle to sleep state. When a job arrives, PM wakes any sleep core up to active state.

We also can save system energy by charging cores not always running at the highest frequency. Frequency controller (FC) regulates the operating frequency according to the predicted workload. The frequency scaling process is based on a discrete time model, with the $T$ as the adjusting interval. We define $w(n)$ as the average workload with frequency $f$ in the interval $(n-1)T \leq t \leq nT$. At end of the interval $nT$ FC predicts the average workload of next interval $w(n+1)$ based on the former average workload. According to the workload predicted FC adjusts the operating frequency of processor core.

4. Simulation results

We use Wattch\(^{(5)}\) to model the switching activity and dynamic power consumption of the multi-core system. The resulting energy consumption (kJ) of none saving policy, DPM, DVS and RESS are shown in the table.

<table>
<thead>
<tr>
<th>Job Set</th>
<th>None (kJ)</th>
<th>DPM (kJ)</th>
<th>DVS (kJ)</th>
<th>RESS (kJ)</th>
<th>ESR (%)</th>
<th>PPR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>345</td>
<td>282</td>
<td>267</td>
<td>235</td>
<td>31.8</td>
<td>2.7</td>
</tr>
<tr>
<td>2</td>
<td>583</td>
<td>456</td>
<td>464</td>
<td>386</td>
<td>33.7</td>
<td>3.1</td>
</tr>
<tr>
<td>3</td>
<td>213</td>
<td>185</td>
<td>187</td>
<td>155</td>
<td>27.2</td>
<td>1.9</td>
</tr>
<tr>
<td>4</td>
<td>426</td>
<td>370</td>
<td>333</td>
<td>297</td>
<td>30.1</td>
<td>3.6</td>
</tr>
</tbody>
</table>

Table 1. Energy consumption and performance hit

The energy saving ratio (ESR) is defined as the energy saved by RESS compared to policy without energy saving. PPR represents the performance penalty ratio which is the ratio of prolonged job execution time for RESS to policy without energy saving. Table 1 shows that system can save 13-22% energy by using DPM method and save 15-24% energy by using DVS at same conditions. We also can see that the usage of RESS achieved 27-35% reduction of energy compared with the non-optimal system. And the system performance penalty of RESS is very small about 1.6-3.9%.

5. Conclusion

In this paper, we present a real-time energy saving Scheduling algorithm to reduce the energy consumption in real-time CMP system. Our algorithm use power manager (PM) to determine the power state of cores. We define a frequency controller (FC) to adopt the operating frequency and voltage dynamically according to the predicted workload on the active cores. The experiment results show that RESS can be used to reduce the system energy cost significantly.

6. Acknowledgment

This research is supported by the National Natural Science Foundation of China under Grant No. 60473098, the Microsoft research Asia fund and The National High Technology Research and Development Program of China No. 2006AA011Z101..

7. References