Optimizing Cloud Data Center Energy Efficiency via Dynamic Prediction of CPU Idle Intervals

Lide Duan\textsuperscript{1} \quad Dongyuan Zhan\textsuperscript{2} \quad Justin Hohnerlein\textsuperscript{1}

lidge.duan@utsa.edu \quad dongyuan.zhan@amd.com \quad csz751@my.utsa.edu

\textsuperscript{1}Department of Electrical and Computer Engineering
University of Texas at San Antonio, San Antonio, TX
\textsuperscript{2}Advanced Micro Devices, Austin, TX

Abstract—The energy consumption of cloud data centers has been growing drastically in recent years. In particular, CPUs are the most power hungry components in the data center. On the one hand, CPUs are not energy proportional with respect to their utilization levels because a cloud server’s energy efficiency is much lower with limited CPU utilizations. On the other hand, current cloud computing applications usually exhibit significant CPU idle time composed of idle intervals of variable lengths. The power consumption in these idle intervals is significant due to the prominent leakage current in recent technology nodes. There are a few existing schemes that transition a CPU into various low-power and sleep states to reduce its idle power. But none of them is optimal due to the fact that entering a sleep state may result in negative power savings if its wake-up latency is longer than the current idle interval. Therefore, intelligent sleep state entry is a key challenge in improving cloud data centers’ CPU energy efficiency. In this work, we propose a dynamic idle interval prediction scheme that can estimate future CPU idle interval lengths and thereby choose the most cost-effective sleep state to minimize power consumption at runtime. Experiments show that our proposed approach can significantly outperform other schemes, achieving 10% - 50% power savings compared to DVFS for a variety of CPU idle patterns.

Keywords—cloud data centers; energy efficiency; CPU idle power; dynamic prediction; sleep states.

I. INTRODUCTION

Cloud computing provides online accesses to computing services and centralized data storage, typically running on data centers where large groups of servers, disks, and routers are networked. Researchers have studied the cost of data centers that provide cloud services, and concluded that about 40% of the total amortized cost of the data center falls into power related categories, including power distribution, cooling costs, electricity utility costs, etc. [9]. Data center energy consumption has been growing drastically in the past several years, nowadays accounting for over 2% of the power usage in the U.S. [13]. As a result, improving energy efficiency has become a first-order consideration in building modern data centers.

Figure 1 demonstrates a peak power consumption breakdown by components in a 2012 Google data center, indicating that a dominant portion of power was drawn by CPUs [4]. This has been a tendency of energy use in the data center evolvement for years. As a result, lowering CPU power consumption should be a primary design goal for data centers. But one key challenge is that CPUs are not energy proportional with respect to their utilization levels [3]: they achieve relatively high performance per watt when they are fully utilized; but the energy efficiency quickly drops with low CPU utilizations. Figure 2 illustrates how the performance and power (on y-axis) would change with respect to different CPU utilization levels (on x-axis). In particular, the power consumption when the CPU is idle (denoted as P(0%)) is above one half of the maximum power (denoted as P(100%)), primarily due to the prominent leakage power in recent technology nodes.

Additionally, contemporary cloud computing applications usually exhibit significant CPU idle time due to their natures of frequent user interactions, insufficient parallelism, etc. [2]. The application appears to be “running”, but the CPU is actually idle. Therefore, a significant amount of energy can be wasted without contributing to useful computation. Clock gating [11] and power gating [2] have been utilized to reduce the CPU idle power via entry into different CPU low-power and sleep states [10]. However, transitioning into and out of these states takes time and consumes power; and much of the CPU idle time is composed of idle intervals of various lengths [2]. In consequence, entering a sleep state may result in negative
power savings if its wake-up latency is longer than the current idle interval. Therefore, intelligent sleep state entry is of vital importance to improving data center CPU energy efficiency.

In this paper, we study the efficacy of a group of existing CPU low-power and sleep state entry schemes. Traditional Dynamic Voltage and Frequency Scaling (DVFS) only transitions a CPU to active but low power states if memory accesses are intense [17-21] in the running workload. A naïve power gating scheme always enters the sleep state as soon as the CPU becomes idle. A known solution in industry [2] always delays the sleep state entry for a fixed amount of time. Although these schemes can achieve certain levels of power savings, their efficacy is still quite limited due to their incapability of adapting to various patterns of CPU utilization and idleness. In order to optimize the power savings under a variety of conditions, we propose a dynamic idle interval prediction scheme that can forecast future CPU idle interval lengths and thereby choose the most cost-effective sleep state to minimize power consumption on the fly. Our proposed approach largely outperforms other schemes examined, achieving 10% - 50% power savings compared to DVFS when using various CPU idle patterns.

II. METHODOLOGY AND ANALYSIS

Current real-world applications experience significant and dynamic idle intervals for many possible reasons [2]: the application typically has an interactive and/or I/O-intensive nature; there is insufficient work for the CPU; there is insufficient parallelism; the introduction of GPUs that offload computation from CPUs; and so on. In a cloud environment, a large number of virtual machines (VMs) run on a smaller number of physical hosts. Multiple VMs receive computing services from the same host via time or space sharing. A VM typically receives a time slice of 30 - 100 milliseconds (ms) from the host before switching to another VM [16]. This time slice is much longer than the CPU idle intervals that usually last from 10 - 500 microseconds (us). Therefore, a lot of CPU idleness can be observed on cloud data center servers.

The CPUs on a data center can transition to low-power or sleep states when their workloads are memory or I/O bounded. Advanced Configuration and Power Interface (ACPI) [1] defines these low-power and sleep states, which have become the standard interfaces implemented in current industrial CPUs. The Intel Xeon processor has the following states [10][12]:

- **C0**: the normal operating state, which includes several P-states that have voltage/frequency scaled to different levels.
- **C1**: the halt state where the clock is stopped.
- **C3**: the sleep state where the cache is flushed.
- **C6**: the deeper sleep state where the architectural state is saved to DRAM and voltage is set to zero.

Each state reduces the power consumption to a different degree and incurs a distinct wake-up latency. Upon idleness, the CPU can choose a state to enter; but entering a state whose wake-up latency is longer than the idle interval length will result in negative power savings and performance degradation. Therefore, we study a set of existing low-power state entry schemes and propose a dynamic prediction solution as follows. For simplicity and without loss of generality, we only consider C0 – C3 in this paper.

- **DVFS**: the traditional Dynamic Voltage and Frequency Scaling, which only operates on the P-states within C0. It is set as the baseline in our study.
- **Naïve Power Gating**: it works like DVFS when CPU is not idle, but always transitions to C3 immediately upon idleness.
- **Fixed Preflush Filtering (FPF)**: it works similarly to Naïve Power Gating, but delays C3 entry for a fixed amount of time. FPF tries to filter out those high frequency and short duration CPU idle intervals. It is productized in AMD’s 15h Trinity processors [2].
- **Dynamic Prediction**: this is our proposed scheme. It dynamically predicts the length of future CPU idle intervals and chooses an appropriate state to transition into, to achieve the best power savings.

Table 1 summarizes the power saving impacts of the low-power state entry schemes above on different types of CPU idle intervals. All comparisons are based on DVFS: “0” indicates no impact; “-” and “+” indicate a negative and a positive impact, respectively. DVFS always operates on C0, regardless of the idle interval lengths. Naïve power gating transitions to C3 immediately upon all idle instances, thus resulting in negative power savings for idle intervals that are shorter than the C3 wake-up latency. FPF filters out short idle intervals, entering C3 only when the idle length is long enough. As a result, it shows a little worse power consumption when the idle length is just slightly longer than the C3 latency, but eventually saves more and more power as the idle length becomes longer. Our dynamic prediction stays in C0 for intervals shorter than the C1 latency, and enters C1 if the interval is longer than the C1 latency but shorter than the C3 delay. For even longer intervals, it compares the power savings between C1 and C3, and chooses the better
one. Details about the new dynamic prediction scheme are described in Section III.

Table 1. The power savings of different low-power state entry schemes w.r.t. distinct types of idle intervals

<table>
<thead>
<tr>
<th></th>
<th>Idle intervals shorter than the C1 latency</th>
<th>Idle intervals longer than the C1 latency but shorter than the C3 latency</th>
<th>Idle intervals longer than the C3 latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVFS</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Naïve</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>FPF</td>
<td>0</td>
<td>0</td>
<td>from - to +</td>
</tr>
<tr>
<td>Prediction</td>
<td>0</td>
<td>+</td>
<td>++</td>
</tr>
</tbody>
</table>

III. DYNAMIC PREDICTION OF CPU IDLE INTERVALS

This section describes the proposed dynamic prediction scheme. We propose to use an advanced statistical model to predict the CPU idle interval lengths based on simple performance measurements, as illustrated in Figure 3. At the architecture level, the variations of a number of CPU performance metrics are monitored, such as IPC, cache miss rates, structure occupancies, branch predictor statistics, etc. Most of these measurements are easy to obtain either from simulators or on hardware via Performance Monitoring Counters (PMCs). We feed the information to a predictor as the inputs. The output of the model is the length of the next CPU idle interval. The model can be trained with widely-used machine learning or statistical techniques, e.g. boosted regression trees [8].

We plan to profile as many performance-related variables as possible, but will identify only a few important ones that the final simplified prediction will be based on. Moreover, the predictor can be trained at runtime, and is capable of making dynamic predictions adapting to the application’s temporal behaviors. Upon accurate predictions, we will be able to dynamically predict how long the next CPU idle interval will last, thus choosing the most cost-effective CPU power state for the upcoming idle interval. This prevents cases such as entering a deep sleep state in a short idle period, thereby avoiding negative power savings and performance penalties due to state transition latencies.

IV. EXPERIMENTS AND RESULTS

Since the dynamic predictor is still an on-going work, we present some preliminary results of our study in this section. Our power model is based on measurements from real systems: the different load levels of C0 use power consumption [5] measured on a HP ProLiant G4 machine (with Intel Xeon 3040 CPU) using the SPECpower benchmark; the other sleep states (C1/C3/C6) use power numbers [14] published by Intel Xeon E5 families. Figure 4 lists the power numbers of different levels used in our DVFS model [5]. Furthermore, we assume a wake-up latency of 10/100/500 microseconds and a power consumption of 47/22/15 Watts [14] for the C1/C3/C6 states, respectively.

We compare the power consumption of the 4 schemes studied in this paper (Section II) based on three types of CPU idle patterns. The “short” pattern consists of idle intervals of 10 to 200 microseconds; the “medium” pattern is comprised of idle intervals of medium lengths ranging from 100 to 500 microseconds; and the “long” pattern consists of idle intervals of 300 to 1000 microseconds. The results are demonstrated in Figure 5, with all power consumption numbers normalized to the corresponding DVFS results in each idle pattern. For demonstration purposes, we use an oracle predictor (“Oracle”) in the last scheme to illustrate the effectiveness of our proposed approach. Nonetheless, a carefully-tuned real predictor usually shows extremely high prediction accuracies [7], achieving a performance level close to that of the oracle predictor.

As expected, Naïve Power Gating incurs significant power overhead (close to 180%) for short idle intervals. This
is because the naïve scheme transitions to C3 regardless of how short the idle interval is, getting negative power savings when the idle length is shorter than C3 wake-up latency. Its performance improves when the idle interval becomes longer, and gains significant power savings for the long idle intervals. For FPF, it avoids the large negative power impact from short intervals by filtering them out with a fixed C3 entry delay (which is set to 200 microseconds in our study). FPF wastes some power for the medium-length intervals whose durations are just a little greater than its C3 transition delay, but eventually saves power as the interval becomes even longer. In contrast, the oracle predictor demonstrates much better power efficiency than the other schemes, and also obtains significant power savings in all idle patterns because of its adaptability. It improves the power consumption of DVFS by 10%/25%/45% for the “short”, “medium”, and “long” idle patterns, respectively. In general, all the C3 entry schemes favor longer idle intervals.

![Figure 5. The power consumption of different schemes (normalized to DVFS) w.r.t. distinct idle interval patterns](image)

V. CONCLUSIONS AND FUTURE WORK

In this work, we study a few CPU low-power and sleep state entry schemes to improve the energy efficiency of cloud data centers. These schemes reduce the power consumption for the variable idle intervals observed on cloud servers. Our proposed dynamic prediction of CPU idle lengths outperforms other heuristics, achieving optimized power consumption for cloud data centers.

Several extensions are being conducted for this work. First, we will develop a practical predictor and compare it against the oracle predictor and other schemes. Different prediction models will be carefully examined and quantitatively studied. Second, we will collect idle traces from real cloud data centers, apply them to our model training, and demonstrate the effectiveness of our new solution. Finally, we plan to incorporate our model into a widely used data center simulation platform, such as CloudSim [6] and BigHouse [15], to provide a detailed and flexible evaluation platform for future studies.

REFERENCES