Power Reduction Through Measurement and Modeling of Users and CPUs: Summary

Bin Lin, Arindam Mallik, Peter A. Dinda, Gokhan Memik, and Robert P. Dick
Department of Electrical Engineering and Computer Science, Northwestern University
Evanston, Illinois, USA
{b-lin, arindam, pdinda, g-memik, dickrp}@northwestern.edu

Categories and Subject Descriptors: D.4 (Operating Systems), C.4 (Performance of Systems), H.5.2 (User Interfaces)

General Terms: Power Management, Human-Computer Interaction, Process Variation

Keywords: Dynamic Voltage and Frequency Scaling (DVFS), Process-Driven Voltage Scaling (PDVS), User-Driven Frequency Scaling (UDFS)

1. INTRODUCTION

Dynamic Voltage and Frequency Scaling (DVFS) is one of the most commonly used power reduction techniques in high-performance processors. DVFS varies the frequency and voltage of a microprocessor in real-time according to processing needs. Although there are different versions of DVFS, at its core DVFS adapts power consumption and performance to the current workload of the CPU. Specifically, existing DVFS techniques in high-performance processors select an operating point (CPU frequency and voltage) based on the utilization of the processor. This approach integrates OS-level control, but such control is pessimistic.

Existing DVFS techniques are pessimistic about the user. Indeed, they ignore the user, assuming that CPU utilization or the OS events prompting it are sufficient proxies. A high CPU utilization simply leads to a high frequency and high voltage, regardless of the user's satisfaction or expectation of performance.

Existing DVFS techniques are pessimistic about the CPU. They assume worst-case manufacturing process variation and operating temperature by basing their policies on loose worst-case bounds given by the processor manufacturer. A voltage level for a given frequency is set such that even the worst shipped processor of a given generation will be stable at the highest specified temperature.

In response to these observations, we have developed, implemented, and evaluated the following two new power management techniques that can be readily employed independently or together. We elaborate on these techniques in detail elsewhere [4].

User-Driven Frequency Scaling (UDFS) uses direct user feedback to drive an online control algorithm that determines the processor frequency. Processor frequency has strong effects on power consumption and temperature, both directly and also indirectly through the need for higher voltages at higher frequencies. The choice of frequency is directly visible to the end-user as it determines the performance he sees. There is considerable variation among users with respect to the satisfactory performance level for a given workload mix [3]. We exploit this variation to customize frequency control policies dynamically to the user. In UDFS, the user presses a button when discomforted by the performance of the machine. These input events drive the UDFS algorithm which sets processor frequency. Unlike previous work (for example, Vertigo [2]), our approach employs direct feedback from the user during ordinary use of the machine.

Process-Driven Voltage Scaling (PDVS) creates a custom mapping from frequency and temperature to the minimum voltage needed for CPU stability, taking advantage of process variation. This mapping is then used online to choose the operating voltage by taking into account the current operating temperature and frequency. Researchers have shown that process variation causes IC speed to vary up to 30% [1]. Using a single supply voltage setting does not exploit this variation. We take advantage of the variation by a customization process that determines the slack of the *individual* processor, as well as its dependence on operating temperature. This offline measurement is then used online to dynamically set supply voltage based on frequency and temperature.

We implemented UDFS as Microsoft Windows client software that appears as a taskbar task. The F11 key serves as the user discomfort button. We developed two algorithms to control the frequency based on these events. The first one (UDFS1) is loosely related to TCP congestion algorithm, the frequency of the processor corresponding to bandwidth and the user input corresponding to packet losses. The second algorithm (UDFS2), on the other hand, assumes that the pressing of a button means that the user wants the processor remain at the level and adjusts itself to remain at that level for a longer time. PDVS is implemented as an offline process (currently based on a remastered Knoppix boot CD and USB flashdrive) that generates a per-processor profile that can then be used by the client software.

2. EVALUATION: SUMMARY+EXAMPLES

Our experiments were done using an IBM Thinkpad T43P with a 2.13 GHz Pentium M-770 CPU and 1 GB memory

This work is in part supported by DOE Award DE-FG02-05ER25691 and NSF Awards IIS-0613568, CNS-0551639, CNS-0347941, CCF-0541337, IIS-0536994, CCF-0444405, ANI-0093221, ANI-0301108, and EIA-0224449. We also like to thank Nikolay Valtchanov and Matthew Robben for their work in implementing the PDVS profiler.

SIGMETRICS'07, June 12–16, 2007, San Diego, California, USA.
ACM 978-1-59593-639-4/07/0006.

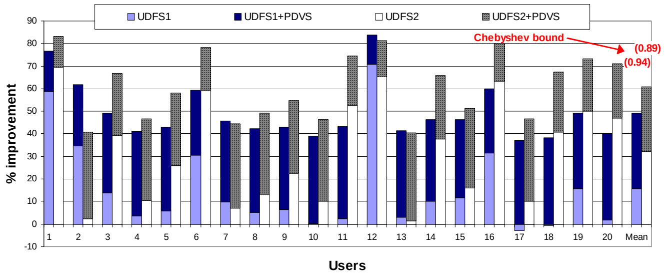

Figure 1: Comparison of UDFS algorithms, UDFS+PDVS, and Windows XP DVFS (CPU Dynamic Power) for the 3D Shockwave animation.

running Microsoft Windows XP Professional SP2. We ran a study with 20 users. The user study took around 45 minutes for each user. Each user was asked perform the following tasks for both UDFS algorithms: Microsoft PowerPoint plus music (4 minutes); 3D Shockwave animation (4 minutes); and FIFA Soccer (8 minutes). Our studies also included multitasking scenarios.

We measure the overall system power and temperature reduction caused by our methods, and derive the CPU dynamic power. Combining PDVS and UDFS schemes reduces measured system power by 49.9% (27.8% PDVS, 22.1% UDFS), averaged across all our users and applications, compared to the Windows XP DVFS scheme. The average temperature of the CPU is decreased by 13.2°C. Using user trace-driven simulation to evaluate the CPU in isolation, we find average CPU dynamic power savings of 57.3% (32.4% PDVS, 24.9% UDFS), with a maximum reduction of 83.4%. In a multitasking environment, the CPU dynamic power is reduced by 75.7% on average. We now present a sampling of our results.

2.1 CPU Dynamic Power

Because we do not have hardware to directly measure CPU power, we collect the time series of CPU frequency over time during the user studies and combine it with the offline PDVS profile (or the nominal voltage settings from the processor datasheet) to derive the CPU dynamic power.

Figure 1 presents both individual user results and average results for both UDFS algorithms, with and without PDVS, for the Shockwave animation. This task elicits the widest range of responses from users, while FIFA Soccer is similar and PowerPoint is much more uniform (and has very high power improvements). Although there is variation from user to user, we reduce power by 55.1% on average. UDFS1 and UDFS2 independently reduce the power consumption by 15.6% and 32.2%, respectively. User 17 with UDFS1 is anomalous. This user wanted the system to perform better than the hardware permitted and thus pressed the button virtually continuously even when the CPU was running at the highest frequency. Adding PDVS lowers average power consumption significantly across the board. The power is reduced by 49.2% (UDFS1+PDVS) and 61.0% (UDFS2+PDVS) in the combined scheme.

2.2 System Power and Temperature

We are able to measure the system power of the laptop by replaying the traces from our user study while the laptop is connected to a National Instruments 6034E data acquisition board attached to a host workstation.

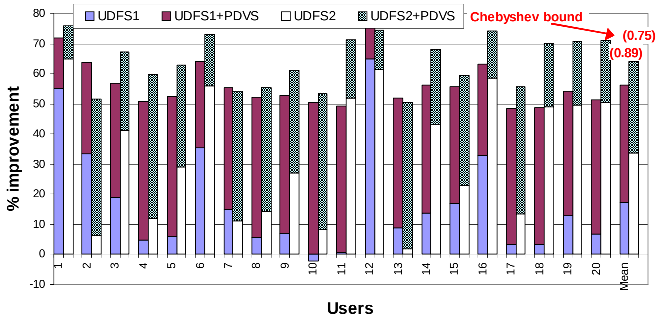

Figure 2: Comparison of UDFS algorithms, UDFS+PDVS, and Windows XP DVFS (Measured system power with display off) for the 3D Shockwave animation.

Figure 2 presents results for the UDFS algorithms with and without PDVS, showing the system power savings over the default Windows DVFS approach for the 3D Shockwave animation. UDFS1 and UDFS2 reduce the average power consumption by 17.2% and 33.6%, respectively. Using UDFS together with PDVS increases average power savings to over 50% over the default Windows DVFS scheme.

3. CONCLUSION

We have identified processor and user pessimism as key factors holding back effective power management for processors with support for DVFS. In response, we have developed and evaluated the following new, process- and user-adaptive DVFS techniques: process-driven voltage scaling (PDVS) and user-driven frequency scaling (UDFS). Extensive user studies show that our techniques result in dramatic power savings over the widely used Windows DVFS scheme. Furthermore, CPU temperatures can be markedly decreased through the use of our techniques. PDVS can be readily used along with any existing frequency scaling approach, while UDFS depends on user feedback.

4. REFERENCES

[1] Borkar, S., Karnik, T., Narendra, S., Tschanz, J., Keshavarzi, A., and De, V. Parameter Variations and Impact on Circuits and Microarchitecture. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC)* (2003).

[2] Flautner, K., and Mudge, T. Vertigo: Automatic Performance-Setting for Linux. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI)* (December 2002).

[3] Gupta, A., Lin, B., and Dinda, P. A. Measuring and Understanding User Comfort with Resource Borrowing. In *Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC 2004)* (June 2004).

[4] Mallik, A., Lin, B., Dinda, P., Memik, G., and Dick, R. Process and user driven dynamic voltage and frequency scaling. Tech. Rep. NWU-EECS-06-11, Department of Electrical Engineering and Computer Science, Northwestern University, August 2006.