A DVFS scheme for Embedded Systems with Human Interaction

Karel De Vogeleer and Gerard Memmi
INFRES – CNRS LTCI - UMR 5141
TELECOM ParisTech
Paris, France
Email: {devogele,memmi}@enst.fr

Pierre Jouvelot and Fabien Coelho
Centre de recherche en informatique
MINES ParisTech
Fontainebleau, France
Email: {jouvelot,coelho}@cri.ensmp.fr

***Abstract*—Dynamic Voltage and Frequency Scaling (DVFS) schemes have shown to be effective in reducing the energy requirements of hand held devices such as smartphones. Current system-wide DVFS schemes act upon statistics of historical performance measures. As a result these schemes react after certain events have manifested. Being proactive to these events may yield more energy and performance gains. We propose a DVFS scheme that bases its decisions on the human interaction with the device. The advantage of such approach is that the system's reactivity is following the computational load on the system better and energy is used more effectively. Our simulations show that such an interactive DVFS scheme is most effective for applications that incur low computational load on the system and interact with user often.**

I. Introduction

The quest for energy optimization in ubiquitous computing is a daunting task. Many optimization techniques originate in the area of High Performance Computing (HPC) and are adapted to function on hand held devices. Moreover, many techniques are (1) application specific and (2) hardware dependent. These two properties are contradictory to the ubiquitousness of today's computing. Hand held devices, for example smartphones and tablets, run on a variety of platforms and run a plethora of applications. Optimization techniques for such devices require to handle bursty performance demands in contrast with HPC systems. Also, most system-wide energy optimization techniques, which are implemented in the Operating System (OS), are based on statistical performance measures. These measures need to be collected before they can be analyzed by the energy-aware decision algorithm. As a result there is a time-lag between an event and the reaction to the event by an energy optimization scheme. In such cases valuable time may be lost to react upon changing operation conditions. A proactive decision scheme would allow the system to be more swift in dynamic energy optimization, and could tentatively also be more adequate in delivering performance when needed.

II. Human interaction based DVFS scheme

We propose a Central Processing Unit (CPU) DVFS scheme for applications that instigates bursty human interaction, e.g, card games, email, web browsing etc. Such applications are idle most of the time and are expected to perform short computations when the user interacts (briefly) with the device. The device must be responsive not to harm the user's experience

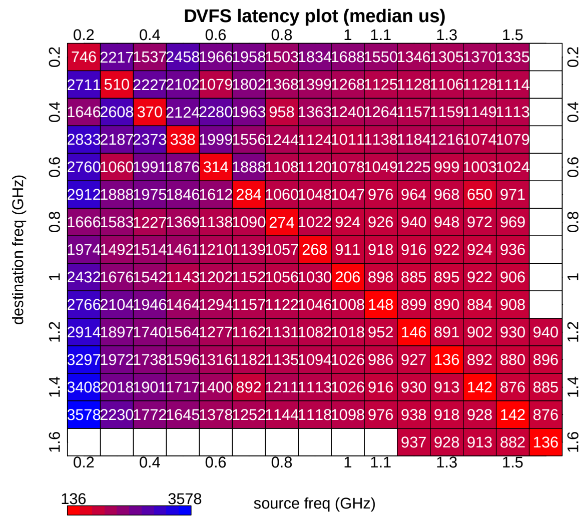

Fig. 1: DVFS transition latency (μs) between CPU frequencies of a Linux powered Samsung Galaxy S2.

while maximizing the idle period of the CPU. Figure 1 shows the transition latency for switching from one CPU frequency to another between 200 MHz to 1.6 GHz of a Linux powered Samsung Galaxy S2. The maximum transition latency shown in Figure 1 is around 3.6 ms. The Galaxy S2 by default only transits between 200 MHz and 1.2 GHz, which have a delay of 2.9 ms for gearing up and 1.3 ms for gearing down. It is important to maintain a real-time stimulus-response in order not to affect the user perception. In professional vehicle simulators this threshold is often assumed not to surpass the 50 ms threshold [1]. Even though this maximum value is pessimistic compared to other DVFS measurements, *e.g.*, Freeh *et al.* [2], frequently changing between CPU frequencies will most likely go unnoticed by the user.

We put forward a DVFS scheme for hand held devices that speeds up when the user interacts with the device and returns to idle when system is not doing any computations related to the foreground applications. This is in contrast with statistical performance based DVFS that will only speed up when the system is loaded for a given timespan and will return to idle when no CPU activity is sensed for a given timespan. The proposed interactive DVFS scheme is anticipated to react faster to human interaction and would provide ondemand performance where the conventional DVFS scheme would lag in time. For the default power management policy of our Galaxy S2, named `ondemand`, the time lag is about 400 ms [3].

In a simplified modeling of reality the timespan of the bursty computation is bounded by (1) the interaction of the user with the device, *e.g.*, a finger touch on the screen, and (2) the end of the computation induced by that user interaction. Our interactive DVFS scheme would gear up during the timespan of said bursty computation. The conventional DVFS scheme would gear up if the CPU is loaded for a given timespan and gear down if it is idle for that a timespan.

We have also shown that the energy/frequency curve of a piece of code may show convex behavior [4]. Measurements pointed out that the CPU intensive benchmark is most energy efficient around 700 MHz. The Galaxy CPU is clocked at 1.2 GHz by default, this implies that running at 700 MHz increases the computation time by 40 %. This time/energy trade-off may be used in the case where the trade-off is justified.

III. Simulation Settings

We wrote a simulation program in `C++` to analyze the trade-off between our human interactive DVFS scheme and the conventional DVFS scheme. We assume that the arrival time of user interaction with a hand held device occurs following a Poisson process with mean λ_a and the triggered CPU computation service time is exponentially distributed with mean λ_s. If the computation following a human interaction is not finished when the next interaction arrives, the computation is queued for processing. The used energy and performance statistics are based on measurements of the Gold-Rader bit-reversal algorithm [4], which is part of the pertinent Fast Fourier Transformation (FFT) algorithm. The conventional DVFS schemes gears between 200 MHz and 1.2 GHz (following the Galaxy S2) and our interactive DVFS scheme gears between 200 MHz and 700 MHz. The bit-reversal algorithm drains 0.100 W, 0.529 W, 1.548 W while computing at 200 MHz, 700 MHz, and 1.2 GHZ respectively. The smartphone's CPU consumes 0.053 W and 0.674 W, when it is idle at 200 MHz and 1.2 GHz respectively.

IV. Performance Study

Figure 2 shows the energy gains relative to the conventional DVFS scheme for different λ_s and λ_a. Values below 1 indicate that the interactive DVFS scheme consumes less energy compared to the conventional scheme. It can be seen that this is the case for inter-arrival times lower than 4 s and service times between 1 s and 5 s. Figure 4 shows the average dead-line delay of the two DVFSs schemes for different λ_s and λ_a. The dead-line delay is defined as the time between the completion of a computation compared to the theoretical completion if the whole system would be available at 1.2 GHz. Values below 1 indicate that the interactive DVFS scheme completes its computational load faster compared to the conventional scheme. It can be seen that for service times below 0.07 s the interactive scheme is most efficient. Small service-times implies short computations.

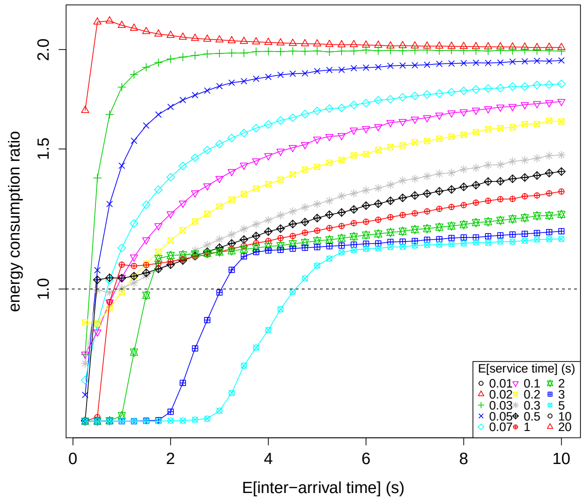

Fig. 3: Energy consumption ratio between the interactive and the conventional DVFS scheme (lower is better).

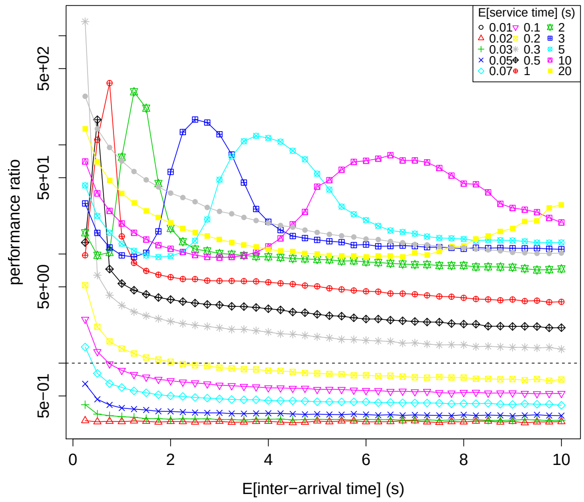

Fig. 4: Performance ratio between the interactive and the conventional DVFS scheme (lower is better).

Both the energy and performance ratio is in favor for the interactive DVFS scheme where the inter-arrival time is between 250 ms and 1250 ms and the computational load is in between 20 ms to 70 ms at 1.2 GHz. This implies that the interactive DVFS scheme performs best for applications where the interaction with the user happens fairly often, in the order of seconds, and where the CPU is lightly loaded. Examples of such applications are card games, log applications, social media applications, image editors etc.

V. Conclusion

We have motivated an interactive DVFS scheme that is adapted to the use on hand held devices. Hand held devices perform usually low computational tasks and are bound by the user interaction. A tailored DVFS scheme that is application aware and proactive to user events could therefore optimize the energy consumption of the hand held device.

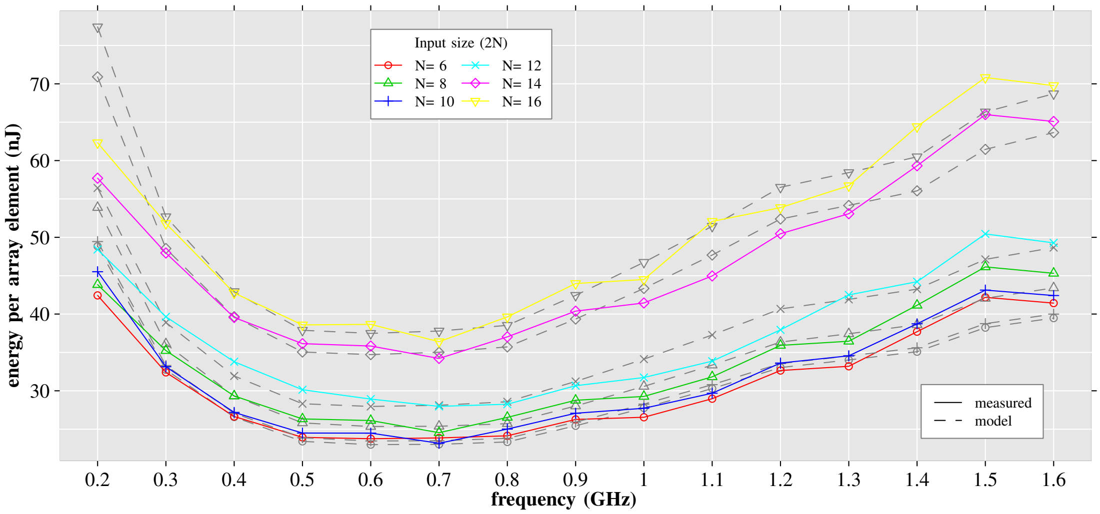

Fig. 2: Energy per array element required by the CPU at 37°C to complete the Gold-Rader bit-reversal algorithm given an input size. The dotted lines denote the theoretical model proposed by the authors [4].

We presented the comparison energy and performance differences between the interactive DVFS scheme and a conventional DVFS scheme based on simulation results. It was shown that the interactive DVFS outperforms that conventional scheme both for energy and performance in the case where the user interaction with the device is in the order seconds and the computational load is low.

References

[1] T. Chapron and J.-P. Colinot, "The new psa peugeot-citroen advanced driving simulator overall design and motion cue algorithm," in *Driving Simulation Conference, North America*, 2007.

[2] V. W. Freeh, D. K. Lowenthal, F. Pan, N. Kappiah, R. Springer, B. L. Rountree, and M. E. Femal, "Analyzing the energy-time trade-off in high-performance computing applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 6, pp. 835–848, Jun. 2007.

[3] S. Kim, H. Kim, J. Kim, J. Lee, and E. Seo, "Empirical analysis of power management schemes for multi-core smartphones," in *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, ser. ICUIMC '13. New York, NY, USA: ACM, 2013, pp. 109:1–109:7. [Online]. Available: http://doi.acm.org/10.1145/2448556.2448665

[4] K. De Vogeleer, G. Memmi, P. Jouvelot, and F. Coelho, "Energy consumption modeling and experimental validation on mobile devices," in *Proceedings on the 10th International Conference on Parallel Processing and Applied Mathematics*. Springer Verlag, September 2013.