New Interfaces for Achieving Power Agility on Mobile Devices

Guru Prasad, Geoffrey Challen
University at Buffalo
{gurupras,challen}@buffalo.edu

Rizwana Begum, Mark Hempstead
Drexel University
{rb639,mhempstead}@drexel.edu

Energy-constrained smartphones are proliferating at an incredible rate. Energy management on these devices, however, remains a major challenge, with consumers citing battery lifetime as their top concern with today's smartphones. As a result, energy-constrained devices are integrating multiple hardware components presenting significant energy-performance tradeoffs. Multiple energy-performance knobs create the potential for applications to choose the right balance of component settings to maintain acceptable performance while saving as much energy as possible. We refer to this ability as *power agility*.

While single component tuning has been studied for nearly a decade [4, 5], operating systems continue to use simple, ineffective approaches that neither coordinate with applications nor isolate energy usage between them [3]. When multiple energy-proportional components are present, they are usually tuned independently, without considering cross-component interactions. As a result, today's operating systems cannot achieve power agility, and Dark Silicon [2] will further expose this deficiency, since including more functionality than can be activated forces the system to make even more energy-performance tradeoffs.

Achieving power agility requires coordination between the application, OS, and hardware: the application understands its performance requirements, the OS must allocate energy between applications over time, and hardware components need guidance about how much energy to consume. These missing interfaces are what prevent today's systems from achieving power agility.

Our architecture introduces the novel interfaces between the application, OS, and hardware, necessary to enable power agility, as well as OS support for translating application guidance into hardware control. Figure 1 provides an overview of the three parts of our system designed to provide power agility. The OS maintains control over energy usage by assigning each application a target *inefficiency*, a novel abstraction that enables energy prioritization while avoiding the pitfalls inherent in bucket-based or rate-limiting approaches. We define inefficiency as the ratio between the amount of energy a task consumes and the minimum amount of energy required to complete the same task. Applications request a balance of component energy-performance tradeoffs by passing a *device description* to the OS. By scaling the device description appropriately to meet the application's inefficiency, the OS can set per-component *energy constraints* which are communicated to hardware.

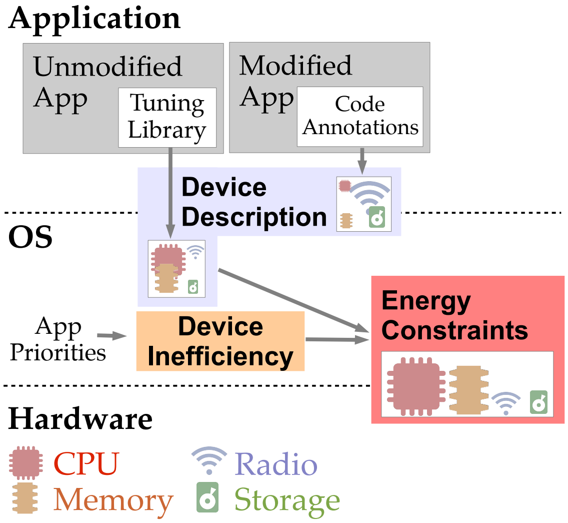

Figure 1: **Our power agile architecture.** The size of the components in the device descriptions represent application requests for cross-component balance.

While power-agile architectures require applications to manage cross-component energy usage, there are multiple ways that they can accomplish this effectively, including adaptive tuning libraries, language support via programmer annotations, and replay based on offline traces.

We use full-system gem5 [1] with an ARM v7 out-of-order processor and the simpleDRAM memory model to experiment with our inefficiency controller. This enables both the processor and memory to be voltage and frequency scaled from 100 MHz to 1 GHz. Our simulations boot Android 4.2 "JellyBean" using Linux version 3.3.0 compiled for gem5.

REFERENCES

[1] Binkert, N., Beckmann, B., Black, G., Reinhardt, S. K., Saidi, A., Basu, A., Hestness, J., Hower, D. R., Krishna, T., Sardashti, S., Sen, R., Sewell, K., Shoaib, M., Vaish, N., Hill, M. D., and Wood, D. A. The gem5 simulator. SIGARCH Comput. Archit. News 39, 2 (Aug. 2011), 1–7.

[2] Esmaeilzadeh, H., Blem, E., Amant, R. S., Sankaralingam, K., and Burger, D. Dark Silicon and the End of Multicore Scaling. In Proceedings of the 38th International Symposium on Computer Architecture (ISCA) (June 2011).

[3] Grover, A. Modern system power management. Queue 1, 7 (Oct. 2003), 66–72.

[4] Isci, C., Buyuktosunoglu, A., Cher, C., Bose, P., and Martonosi, M. Phases: Duration Predictions and Applications to DVFS. IEEE Micro (2005).

[5] Isci, C., Buyuktosunoglu, A., Cher, C., Bose, P., and Martonosi, M. An Analysis of Efficient Multi-Core Global Power Management Policies: Maximizing Performance for a Given Power Budget. In International Symposium on Microarchitecture (MICRO) (December 2006).