

E-MOS: Efficient Energy Management Policies in Operating Systems

Vinay Gangadhar
vinay@cs.wisc.edu

Clint Lestourgeon
clint@cs.wisc.edu

Jay Yang
jkyang2@wisc.edu

Varun Dattatreya
channagirida@wisc.edu

1. Related Work

Energy Management (EM) area in operating systems is full of diverse set of solutions to the problem of ensuring performance while decreasing power usage. The Linux kernel, has some predefined governors for managing power use, but many papers have recognized that these general purpose solutions, are certainly not optimal, and perhaps in some case not even adequate. There have been a number of attempts to remedy this with solutions enlisted here. Dynamic Voltage Frequency scaling [1] has been a popular mechanism to save power and boost performance. Zeng et. al [7] focuses on directly managing power usage as one might manage other scarce resources. Choi et. al [2] dynamically profiles applications to determine a good energy management policy at hardware level.

Non-Optimal General Purpose EM Policies It has been recognized that the generic policies for energy management implemented in Linux are not always the best performing, or most efficient. For example in the context of mobile devices [4], the on-demand power governor used by the Android operating system does not guarantee low power consumption for all workloads. On the opposite end of the spectrum, we see work done by Yapei. et. al [5] that attempts to optimize for the case of a datacenter and provide QOS guarantees more rather than energy efficiency. So, our work focuses on providing more dynamic energy management decisions based on the application information at kernel level. Our work is similar to the work in [4], in the sense that we plan on using the User Space power governor that is exposed to user applications to drive our power policy and comparing our implementation with the other Linux governors.

Existing Energy Efficiency Policies In the case of [7], energy is treated as another scarce resource, and a currency based algorithm is used. Processes in this system are allocated a specific amount of power usage, and are given access to various power consuming operations in relation to the amount of power they are allocated. However, this is focused largely on the question of optimization of allocation, and not on the question of how to ensure that the energy is productively used. Other implementations [4, 2] attempt to form a correlation between the number of memory accesses and ideal operating frequency. They base this argument on the fact that some workloads see lower power when executed at higher frequencies. Their experiments show that this is a consequence of the number of memory accesses, whose frequency of operation remains constant. By getting a measure of the memory accesses in a workload, they use the memory access - CPU frequency correlation to set the operating frequency. For our policy, we plan on classifying workloads depending on how CPU bound

they are or on how much time they spend waiting on external processes. We could then determine suitable power governor policies for these classified workloads.

Application Profiling and decomposition Dynamic voltage and frequency scaling based on application profiling and decomposition has been done in Choi. et. al [2]. This idea on profiling is similar to our project's initial profiling step, but the workload is decomposed into two-parts here: On-chip and Off-chip, effectively indicating CPU sensitive and memory sensitive applications respectively. It exploits on the idea that different workloads have different power management needs and this statistics could be exploited by the performance monitoring unit (PMU) at run-time. They do not implement this in operating system and overall energy management policy has to still go through kernel and the kernel may decide to override the policy based on its decisions. Similarly, Choi et. al [3] work does DVFS for energy conservation classifying all the applications based on On-chip and Off-Chip computation ratio. Again this would result in a generic decision, based on ratios calculated offline. Power conscious fixed scheduling policy of applications with DVFS based on profiling information has been implemented in Shin et. al [6], which again does not integrate the decision making capability to operating systems. Our project aims at embedding this application information into energy management decision making policy of operating system kernel and does this based on application needs.

References

- [1] R. Z. Ayoub, U. Ogras, E. Gorbato, Y. Jin, T. Kam, P. Diefenbaugh, and T. Rosing, "Os-level power minimization under tight performance constraints in general purpose systems," in *Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design*. IEEE Press, 2011, pp. 321–326.
- [2] K. Choi, R. Soma, and M. Pedram, "Dynamic voltage and frequency scaling based on workload decomposition," in *Proceedings of the 2004 international symposium on Low power electronics and design*. ACM, 2004, pp. 174–179.
- [3] —, "Fine-grained dynamic voltage and frequency scaling for precise energy and performance tradeoff based on the ratio of off-chip access to on-chip computation times," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 24, no. 1, pp. 18–28, 2005.
- [4] Y. Liang, P. Lai, and C. Chiou, "An energy conservation dvfs algorithm for the android operating system," *Journal of Convergence*, vol. 1, no. 1, 2010.
- [5] Y. Liu, S. C. Draper, and N. S. Kim, "Sleepscale: runtime joint speed scaling and sleep states management for power efficient data centers," in *Computer Architecture (ISCA), 2014 ACM/IEEE 41st International Symposium on*. IEEE, 2014, pp. 313–324.
- [6] Y. Shin and K. Choi, "Power conscious fixed priority scheduling for hard real-time systems," in *Design Automation Conference, 1999. Proceedings. 36th*. IEEE, 1999, pp. 134–139.
- [7] H. Zeng, C. S. Ellis, A. R. Lebeck, and A. Vahdat, "Ecosystem: Managing energy as a first class operating system resource," *ACM SIGPLAN Notices*, vol. 37, no. 10, pp. 123–132, 2002.