POSTER – Firestorm: Operating Systems for Power-Constrained Architectures

Sankaralingam Panneerselvam
University of Wisconsin-Madison
sankarp@cs.wisc.edu

Michael Swift
University of Wisconsin-Madison
swift@cs.wisc.edu

Hardware Trends. Moore's law paved the way for doubling the transistors in the same chip area with every generation. However, with the end of Dennard's scaling, voltage and hence the power draw of transistors is no longer dropping proportionally to size. As a result, modern processors cannot use all parts of the processor simultaneously without exceeding the power limit. This manifests as an increasing proportion of *dark silicon* [4]. In other words, the compute capacity of current and future processors is and will be over-provisioned with respect to the available power.

Power limits are influenced by different factors such as the capacity of power distribution infrastructure, battery supply limits, and the thermal capacity of the system. Power limits in datacenters can arise from underprovisioning power distribution units relative to peak power draw. Energy limits are also dictated by the limited capacity of batteries. However, in many systems, the primary limit comes not from the ability to acquire power, but instead from the ability to dissipate power as *heat* once it has been used.

Thermal limits are dictated by the physical properties of the processor materials and also comfort of the user. Thus, power is limited to prevent processor chips from overheating, which can lead to thermal breakdown. As a result, the maximum performance of a system is limited by its *cooling capacity*, which determines its ability to dissipate heat. Cooling capacity varies across the computing landscape, from servers with external chilled air to desktops with large fans to laptops to fan-less mobile devices.

Current Support. Processors support mechanisms to enforce both power and temperature limits. For example, recent Intel processors provide Running Average Power Limit (RAPL) counters to enforce a power limit on the entire processor [1]. In software, power capping services, such as the Linux power capping framework uses these limits to control power usage. Processor vendors define Thermal Design Power (TDP) metric for every processor model to guide the requirements of the cooling system needed to dissipate power. Most processors have a safeguard mechanism that throttles the processor by reducing the frequency or duty

cycle (fraction of cycles where work happens) on reaching a critical temperature. In software, the thermal daemon [11] tries to achieve system-wide goals, such as maximum performance by increasing cooling before throttling the processors.

Challenges. The drawback with current hardware and software mechanisms that enforce power and thermal limits are that *they only offer system-wide guarantees but are ignorant of individual application needs and priorities*. This raises following challenges in a shared environment.

Power distribution: When power is limited, current systems throttle all applications equally. However, this approach ignores users' scheduling priorities: power should be distributed among applications based on their importance, so that high-priority applications can use more power to run faster, while low-priority applications have their power reduced and bear most of the performance lost.

Thermal interference: An application that makes heavy use of a processor can trigger temperature throttling that reduces CPU frequency or duty cycle. This can affect all cores, and hence all running applications. As a result, a low priority or malicious application can trigger throttling that reduces the performance of high-priority applications.

Performance Boosting: Applications may be able to temporarily overclock the core to achieve extra performance if they run for short enough periods such that throttling does not get triggered. This is often called computational sprinting [9]. Turbo Boost [6] is similar but more conservative. The challenge is to ensure sufficient thermal headroom to run without overheating by not allowing other applications to exhaust the thermal capacity of the processor.

Firestorm. Just as the OS actively manages resources such as CPU and memory, we argue that operating systems should treat power and thermal capacity as primary resources in the system. Towards this goal, we built Firestorm, an extension to Linux that adds power and thermal awareness. Firestorm introduces new abstractions to enable resource management over power and thermal capacity. It presents new interfaces for applications to gather power and thermal resources, and allow application-specific use of power and thermal capacity for guaranteed performance. Finally, it employs new policies to balance the varied performance requirements of different applications. The overall architecture of Firestorm is shown in Figure 1.

Power as a Resource. Firestorm focuses on power-limited architectures where all compute units cannot be used at maximum performance within the power budget. Firestorm enforces control over the total power consumed by processing

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

PACT '16 September 11-15, 2016, Haifa, Israel

© 2016 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-4121-9/16/09.

DOI: <http://dx.doi.org/10.1145/2967938.2975607>

1. REFERENCES

- [1] “Intel[®] 64 and IA-32 Architectures Software Developer Manuals,” <http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html>, December 2015.
- [2] AMD Corporation, “AMD Turbo Core Technology.”
- [3] Andrei Frumusanu, “Early Exynos 8890 Impressions And Full Specifications,” <http://www.anandtech.com/show/10075/early-exynos-8890-impressions>.
- [4] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, “Dark Silicon and the End of Multicore Scaling,” in *Proc. ISCA*, Jun. 2011.
- [5] M. Goma, M. D. Powell, and T. N. Vijaykumar, “Heat-and-run: Leveraging smt and cmp to manage power density through the operating system,” in *Proceedings of the 11th International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS XI, 2004, pp. 260–270.
- [6] Intel Corporation, “Intel Turbo Boost Technology 2.0,” <http://www.intel.com/content/www/us/en/architecture-and-technology/turbo-boost/turbo-boost-technology.html>.
- [7] J. Li and J. F. Martinez, “Dynamic power-performance adaptation of parallel computation on chip multiprocessors,” in *High-Performance Computer Architecture, 2006. The Twelfth International Symposium on*, Feb 2006, pp. 77–87.
- [8] S. Panneerselvam and M. M. Swift, “Chameleon: operating system support for dynamic processors,” in *Proceedings of the seventeenth international conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS ’12, 2012, pp. 99–110.
- [9] A. Raghavan, Y. Luo, A. Chandawalla, M. Papaefthymiou, K. P. Pipe, T. F. Wenisch, and M. M. K. Martin, “Computational sprinting,” in *Proc. 18th HPCA*. Washington, DC, USA: IEEE Computer Society, 2012, pp. 1–12.
- [10] T. Somu Muthukaruppan, A. Pathania, and T. Mitra, “Price theory based power management for heterogeneous multi-cores,” in *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS ’14, 2014, pp. 161–176.
- [11] “Linux Thermal Daemon,” <https://01.org/linux-thermal-daemon>.