2020 IEEE/ACM International Workshop on Runtime and Operating Systems for Supercomputers (ROSS)

Pinpoint the Joules: Unifying Runtime-Support for Energy Measurements on Heterogeneous Systems

Sven Köhler[1], Benedict Herzog[2], Timo Hönig[3], Lukas Wenzel[1],
Max Plauth[1], Jörg Nolte[4], Andreas Polze[1] and Wolfgang Schröder-Preikschat[2]

[1] Hasso Plattner Institute for Digital Engineering, University of Potsdam, Germany
[2] Friedrich-Alexander University Erlangen-Nürnberg (FAU), Germany
[3] Ruhr University Bochum (RUB), Germany
[4] Brandenburg University of Technology Cottbus–Senftenberg (BTU), Germany
{firstname.lastname}@{hpi,cs.fau,rub,b-tu}.de

***Abstract*—For the design and operation of today's computer systems, power and energy requirements are highest priorities. Unlike performance analyses, however, power and energy measurements of heterogeneous systems are difficult to conduct. Especially at the system-software level, performing power and energy measurements remains challenging. Yet, such measurements are essential to improve software components for low power and high energy-efficiency.**

In this paper, we analyze and discuss the power and energy characteristics of several heterogeneous systems with up to 20 cores (160 hardware threads) and 1 TB of main memory. For the analyzed systems, we outline challenges regarding power and energy measurements and show ways to overcome limitations (i.e., sampling constraints). To improve the current state of the art in power and energy measurements at the system-software level, we present the design and implementation of PINPOINT, an energy-profiling tool which unifies different power and energy measurement interfaces.

***Index Terms*—Power and Energy Measurements, Heterogeneous Systems, System Software**

I. Introduction

Over the past years, power and energy demand have become primary design criteria and critical operating resources for computing systems. This applies to all types of systems, from embedded systems to desktop computer systems, high-performance computers, and supercomputers.

Until recently, the focus of system development has been on performance. However, as technologies have progressed (i.e., parallel computing on multi- and many-core processors) to address technologic limitations (i.e., breakdown of Dennard Scaling, pervasion of dark silicon [1]), power and energy requirements became the crucial points in computer systems' design. Due to the diversity of different computing system types (and their varying purposes), power and energy are important for various reasons. For small and embedded systems available power and energy resources must be handled with care to maximize the operating time (i.e., battery life) of the computing systems [2], [3]. This is due to the limited progress of battery technologies and the mediocre increase of energy density of energy storage over the last decades [4]. For large, high performance and cluster computing systems, however, power and energy became limiting factors as to thermal stress and external constraints (i.e., dependencies at the grid level) [5], [6]. The current Top-500 #1 system *Fugaku* can demand over 28 MW peak power [7], which easily exceeds the power demand of entire towns. The corresponding challenges arise by run-time requirements which demand dynamic adaptation of system characteristics (i.e., change between low and high power operations) *on the go* [8].

To improve the performance of computing systems, considerations about the system *performance* [9], [10] commonly is measured in the time dimension. For example, by specifying the execution time that is required to complete a single operation or a compound task. In the network domain, *throughput* [11] (e.g., rate at which operations are completed within a period) and *latency* [12] (e.g., delay of processing time of certain operations) are also seen on the background of *time*. For different types of systems, performance analysis procedures are similar: hardware-level timers are exposed via the operating system as programming interfaces. Software developers can use these facilities (Linux high resolution timers, for example) accordingly to analyze and improve performance, throughput, and latency of their program code at run-time.

However, there are no alike, standardized programming interfaces available in order to analyze or improve *power* and *energy* characteristics of computing systems. Such easy-to-use interfaces would be especially important as energy and power efficiency not always correlate with performance [13]. Instead, each and every hardware platform provides different means (if any) to obtain power values during run-time. As a result, there are no unified methods of performing measurements to determine the system power demand for the majority of platforms. Power measurements in combination with time measurements can be used to reflect about the energy demand (power over a period of time) of a system [14]. Thus, energy demand analyses of computing systems suffer from the same fact that unified programming interfaces are unavailable.

The increasing amount of computing systems' heterogeneity (i.e., CPUs, GPUs, TPUs[1]) make the situation even more

[1] tensor processing unit, accelerator application-specific integrated circuits which improve the execution of neural networks

978-1-6654-2268-0/20/$31.00 ©2020 IEEE

DOI 10.1109/ROSS51935.2020.00009