Approximation for Run-time Power Management

Anil Kanduri[1], Mohammad-Hashem Haghbayan[1], Amir M. Rahmani[2,3], and Pasi Liljeberg[1]
[1]University of Turku, Finland, [2]University of California, Irvine, USA, [3]TU Wien, Austria
{spakan, mohhag, pakrli}@utu.fi, amirr1@uci.edu

***Abstract*—Performance and energy efficiency of multi-core and many-core systems are restricted by increasing power densities and/or limited energy resources. Maximizing performance while minimizing power and energy consumption becomes challenging with emerging workloads. Approximate computing is an alternative solution that offers the required performance and energy gains, leveraging inherent error resilience of specific application domains. Dynamic power management using approximation as another knob can maximize performance and energy efficiency within fixed power budgets. Disciplined tuning of approximation along with other traditional power knobs requires efficient run-time resource management techniques. We present our strategy for using approximation as another knob for tuning the performance loss incurred in power actuation in many-core systems, which is also portable for heterogeneous multi-core systems.**

I. Introduction

Power densities of multi-core and many-core systems are increasing rapidly with aggressive transistor scaling. To avoid the consequent thermal violation, the system has to function within a fixed budget of power, restricting the performance and energy efficiency. The inactivity forced due to the thermal constraints is termed as Dark Silicon, reflective of the amount of simultaneously utilizable resources of a chip [4]. Multi-core systems that are battery powered are further restricted by the limited power and energy resources available [10]. Heterogeneous architectures [10], dark silicon aware resource management [11], custom hardware acceleration [8] have been proposed to meet performance and energy efficiency challenges. With emerging class of machine learning applications spanning over artificial intelligence, computer vision, Internet-of-Things (IoT) domains, real-time performance and energy efficiency requirements continue to grow [8]. Traditional power management techniques are limited in terms of maximizing the performance within fixed power and energy budgets [9]. On the verge, approximate computing has emerged as one of the possible solutions to satisfy the performance and energy demands.

Approximate computing trades off accuracy for performance and energy gains, relying on applications' inherent error resilience. Applications from widely used domains such as multi-media processing, computer graphics and vision, animation, gaming and learning exhibit a degree of tolerance to inaccurate computations. These applications often operate on redundant or noisy data, iterative algorithms, and have human perception as possible end result. Error resilient nature of these applications can be leveraged to relax the workloads - to meet the performance and energy efficiency requirements, achieved within an acceptable loss of accuracy. Figure 1 shows the performance gains for different machine learning kernels with the amount of workload relaxed. The amount of error induced in each of these cases depends largely on input data. The applications used viz., linear regression, least squares curve fitting, k-Means clustering and k-nearest neighboring classification rely on iterative algorithms such that the computations converge towards an optimal solution. Relaxing some of the input data and/or convergence requirements on such kernels reduces the workload significantly, thus providing an improved performance within similar power consumption. This scenario can be further explored to achieve either a high performance and high energy efficient operation or a nominal performance and low power and thermal operation.

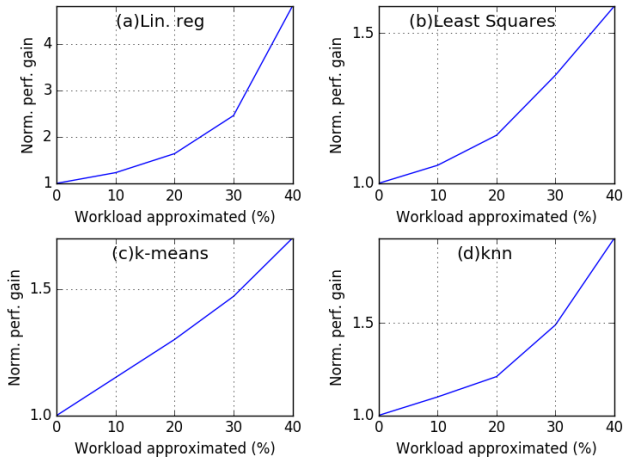

Figure 1. Performance gains with approximation. (a) Linear regression, (b). Least squares curve fitting, (c). k-means clustering, (d). knn classification. Workload approximated corresponds to percentage of loops skipped within each kernel.

With dynamic workload characteristics, variable system's power-performance characteristics over different architectures, designing scalable solutions for exploiting approximation at run-time is challenging. Combining approximation with other traditional power and performance knobs requires complex run-time management solutions. We describe our run-time strategies exploring approximation (APPX) as another knob for power capping and energy efficiency and its adaptability over different architectures.

II. Approximation Knob

Computer systems operate under fixed power budget, thermal design power (TDP), to stay within safe thermal limits [1]

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

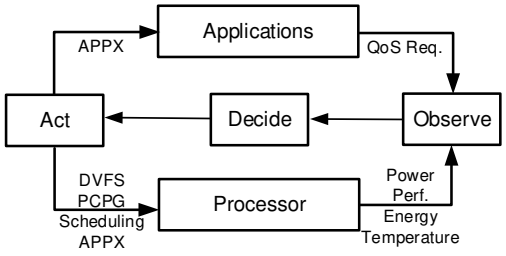

Figure 2. Approximation as a power/performance knob combined with traditional power knobs

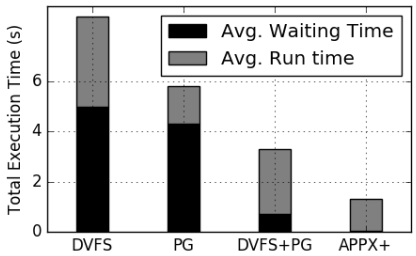

Figure 3. Performance gains with approximation knob combined with power knobs versus individual power knobs alone

[7]. To cap the power consumption below TDP, power knobs such as voltage and frequency scaling (DVFS), power and clock gating, core folding etc., are dynamically triggered [3]. Run-time power and thermal management strategies rely on these traditional power knobs for power capping and actuation. Typical power capping techniques function in an observe-decide-act loop, monitoring critical chip parameters, making power and other resource allocation decisions, followed by enforcement of the decisions made, as shown in Figure 2. Although power capping techniques are effective in restricting power consumption to a safer limit, reduction in such resources degrades performance. Approximation as a knob can complement the other power knobs to ensure no loss in performance while also improving the energy efficiency. This requires hierarchical control and actuation of APPX along with DVFS and power gating (PG), to maximize performance within fixed power budgets, which in turn can improve both per-application performance and overall chip throughput. Figure 2 shows the combination of traditional power knobs along with APPX knob, that interacts with application - for software approximation and architecture - for hardware approximation. It should be noted that the traditional power knobs are reactive in nature, with relatively smaller overhead and satisfy short-term goals. In contrast, APPX is to be triggered pro-actively with some overhead in invocation, however yields larger benefits in long term. With reduced workloads allowing reduction in system resources, APPX can be used opportunistically to achieve i) low power operation offering same performance and ii) high performance operation within same power consumption.

The performance of traditional power knob combinations with and without approximation knobs is shown in Figure 3. We simulated sparse matrix multiplication over cycle accurate many-core platform [5] (described in the following subsections) with power management using DVFS, PG and the combinations of DVFS+PG, and DVFS+PG+APPX (shown as APPX+). We define average waiting time (AWT) as the time elapsed between arrival of an application and beginning of its execution. With insufficient power budgets and/or limited free cores available on the chip for execution typically results in longer AWT. We present performance measures a combination of AWT and original execution time. It is evident that performance with APPX knob in combination with other power knobs is higher due to the reduced workloads. The improvement in per-application latency in turn also provides free cores for incoming applications, reflective of the negligible AWT. With this motivation, we propose to integrate APPX as another knob within the control structure of run-time power management frameworks.

A. Run-time Mode Switching

The idea of exploiting approximation at run-time can be achieved by switching the mode of execution of an application or a sub-task within an application from accurate to approximate. Upon a resource management emergency event i.e., violation of power budget or high performance requirement from dynamic workloads, we trigger the approximation knob while traditional power knobs continue to act on power actuation. Mode switching is analogous to task migration in heterogeneous systems, where a specific task is migrated onto specialized cores - instead of physically migrating the task, we replace an accurate task with its approximate version on the same core. We consider a power violation as power consumption exceeding the fixed power budget of TDP. We consider AWT exceeding a fixed threshold (AWT_{th}) as a performance requirement event. In either cases, we use mode switching to replace the accurate task with its approximate version. The hierarchical view of our resource management platform is shown in Figure 4.

Incoming applications are serviced by the run-time mapping unit by finding appropriate free cores on the chip. For each application, tasks that are approximable are stored in the task bank, to be reused if mode switching becomes necessary. Load analyzer monitors incoming applications' request rate, which is measured as AWT. This value is compared against the pre-determined threshold AWT_{th} to monitor performance requirement event. Power consumption of the chip is monitored through on-chip sensors to determine power budget available. Voltage and frequency settings proportional to the available power budget are set using a PID controller. Both power and performance metrics are considered by the Knob setting which invokes DVFS, PG and APPX knobs as per the application requirements and system's power dynamics. Voltage and frequency settings and power gating are directly communicated to the voltage regulators on the chip. Invocation of approximation is handled by the run-time mapping unit, which retrieves the approximate version of the task specified

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

by Knob settings from the task bank where the approximate version is previously stored. The mapper replaces the accurate version currently running on the chip with the approximated version from the task bank. After mitigating the power and/or performance emergency, accurate version of the task is restored again from the task bank and is replaced with the approximate versions running, if any.

B. Implementation

We implement our mode switching technique and the run-time management policy over many-core architecture, described in [11]. Each processing element on the many-core system is modeled as per Niagara-2 like in order cores and inter-core communication is provided by Noxim infrastructure [5]. We simulate the workloads on an in-house cycle accurate simulator, which is an extension to Noxim framework [6]. Applications are modeled as task graphs representing computational intensity and inter-task communication volume. Loop perforation and relaxed convergence are used to obtain approximate versions of specific tasks. The computation and communication volumes necessary for modeling the task graphs are extracted from Sniper simulator [2]. Technology scaling and power models are obtained from Lumos framework [13]. The system is equipped with per-core DVFS, power gating and run-time mapping support. We evaluated our proposed approach APPX+ as a combination of APPX, DVFS and PG against the PG and DVFS knobs alone and the combination of PG+DVFS. We used workloads with a combination of 4 applications from machine learning domain including linear regression, least squares, k-means clustering and k-nearest neighbors classification. We transformed the applications into task graphs structures and we used two levels of approximate version for each application. For evaluation purpose, we simulated the system for 100 applications of above combinations to enter and leave the system. Figure 5 shows the normalized per-chip throughput of our approach versus other power knob combinations. Performance is progressively higher with PG, DVFS, PG+DVFS and APPX+PG+DVFS respectively. APPX+ offers highest throughput which is 1.5 $\times$ better than the traditional power knobs alone. The error induced with approximation ranged between 3-11 %, which is tolerable given the resilient nature of the applications.

III. Approximation for Run-time Management in Multi-core Systems

Heterogeneous multi-core systems comprising of cores with diverse power-performance characteristics present a challenging scenario for extracting high performance within fixed power budget. Since they are battery-powered and run a variable workloads, energy resources and thermal constraints are stringent. While power actuation degrades per-chip throughput in many-core systems, it also specifically effects per-application latency in the case of multi-core systems. Previously, we explored approximation inclined as a performance knob, although reducing workloads with approximation and compounding it with traditional power knobs can result in

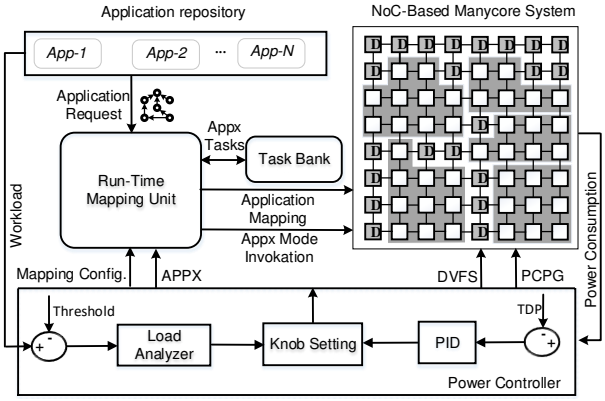

Figure 4. System architecture of power management framework [9]

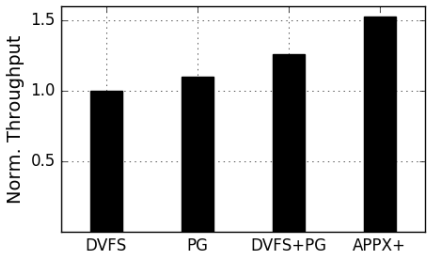

Figure 5. Normalized throughput using different knobs

power reduction too. We present a motivational example of k-means clustering application with and without approximation on a heterogeneous platform. For evaluation, we used Odroid XU3 which consists of 4 ARM A15 (big) cores and 4 ARM A7 (LITTLE) cores. We collect power consumption of the application run on big cluster from the on board power sensors. We used loop perforation to reduce workloads, to realize approximate execution [12]. Figure 6 shows the accuracy-power trade-offs with approximation (APPX) knob, used in combination with traditional power knobs viz., (a) DVFS and (b) CPU Utilization. Reduced workloads with APPX knob invocation allows lowering of voltage and frequency levels (DVFS) and/or CPU Utilization for low power operation, while providing required performance. Higher percentage of loops skipped allows further lowering of CPU resources, resulting in higher power savings. Conversely, using approximation without lowering DVFS or CPU Utilization can provide higher performance within the same power consumption. This establishes the baseline for employing approximation as both performance and power knob, to achieve better performance-per-watt.

In the previous section, we presented approximation as a knob for homogeneous many-core systems. The same principle can be extended for heterogeneous multi-core systems, where power actuation and performance management is critical due to limited power and energy resources. However, combining approximation with traditional power knobs becomes complex optimization problem with heterogeneous systems with avail-

978-1-5386-4881-0/18/$31.00 ©2018 IEEE

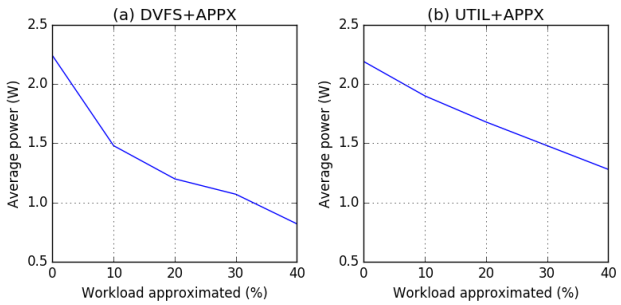

Figure 6. Power gain with approximation. (a). Combination of DVFS and APPX, (b). Combination of CPU Utilization and APPX. Workload approximated is percentage of loops skipped within a kMeans application

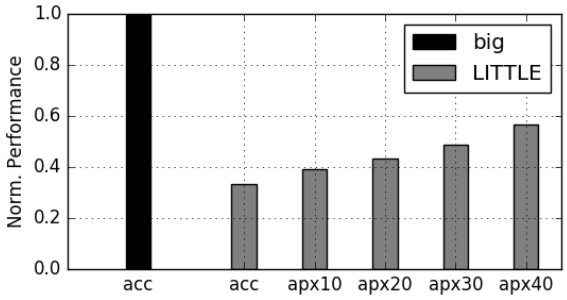

Figure 7. Implication of approximation combined with task migration on performance. acc-accurate exceution, apx10 to apx40 - approximate versions with 10 to 40 % of workload reduced

ability of task migration and CPU utilization control knobs. The effect of task migration in particular varies over different workloads, making resource allocation less trivial.

Exploiting Heterogeneity: With concurrent and dynamic workloads, resource allocation decisions combining task migration, CPU utilization, DVFS and approximation requires coordinated and hierarchical control system for knob actuation decisions. Specific to big.LITTLE like architectures with a combination of high performance- power hungry and low performance-power conserving cores, migration among clusters alters power-performance dynamics. Despite low power consumption, migrating from big cores to LITTLE cores causes significant performance degradation. Power actuation of traditional knobs along with task migration thus becomes limited in terms of offering required performance within power constraints. Combining approximation with task migration can minimize the performance penalty, thus compromising on accuracy instead of performance. We present a trivial combination of approximation with task migration, which can preserve performance guarantees upon migration from big to LITTLE cores. Figure 7 shows performance of the same example k-means application presented in Figure 6 on big and LITTLE clusters, with and without approximation. The accurate version of the application suffers almost 3 $\times$ degradation in performance when migrated from the big cluster to the LITTLE, although power consumption is reduced. Invoking approximation upon task migration minimizes the performance loss with execution on LITTLE cores, while power benefits from task migration are already achieved. With further relaxation on accuracy requirements, performance of approximate version on LITTLE cluster is almost 60% of that on the big cluster. Based on the examples presented, invocation of approximation requires fine tuning to set appropriate level of accuracy to maximize performance and minimize power consumption. Under dynamic workloads running on heterogeneous architectures, finding the right combination of approximation along with DVFS, CPU Utilization, scheduling and task migration becomes far challenging. Efficient run-time strategies however can combine approximation with other power knobs to meet both application's performance requirements and system's power constraints.

IV. Conclusions and Future Work

Power actuation in computer systems based on traditional power knobs often degrades performance of applications for thermal safety. Approximate computing has emerged as an alternative for extracting high performance within low power. With this motivation, we propose approximation as a dynamic performance tuning knob, to be used in run-time management along with traditional power knobs. We presented our power management framework that uses the combination of DVFS, PG and APPX to cover up for the any performance lost in power actuation. We evaluated our approach over machine learning kernels that are inherently error resilient. We present the idea of extending our framework into heterogeneous multi-core system which involves a complex scenario of task migration and core level utilization. Implementation of our run-time manager as operating system level policy for heterogeneous systems is planned for future work.

References

[1] Intel Corporation. Intel Xeon Processor - Measuring Processor Power, revision 1.1. In *White paper, Intel Corporation*, 2011.
[2] Trevor E. Carlson, Wim Heirman, and Lieven Eeckhout. Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations. In *SC*, 2011.
[3] R. Cochran et al. Pack & cap: adaptive dvfs and thread packing under power caps. In *MICRO*, 2011.
[4] H. Esmaeilzadeh et al. Dark silicon and the end of multicore scaling. In *ISCA*, 2011.
[5] F. Fazzino et al. Noxim: Network-on-chip simulator. *URL: http://sourceforge.net/projects/noxim*, 2008.
[6] M.-H. Haghbayan et al. Dark Silicon Aware Power Management for Manycore Systems under Dynamic Workloads. In *ICCD*, 2014.
[7] S. Pagani et al. TSP: Thermal Safe Power: Efficient Power Budgeting for many-core systems in dark silicon era. In *CODES+ISSS*, 2014.
[8] Norman P Jouppi et al. In-datacenter performance analysis of a tensor processing unit. *arXiv preprint arXiv:1704.04760*, 2017.
[9] Anil Kanduri et al. Approximation knob: Power capping meets energy efficiency. In *In Proc. of ICCAD*, pages 1–8. IEEE, 2016.
[10] T.S. Muthukaruppan et al. Hierarchical power management for asymmetric multi-core in dark silicon era. In *DAC*, 2013.
[11] A. Rahmani et al. Dynamic power management for many-core platforms in the dark silicon era: A multi-objective control approach. In *ISLPED*, 2015.
[12] S. Sidiroglou et al. Managing performance vs. accuracy trade-offs with loop perforation. In *FSE*, 2011.
[13] L. Wang and K. Skadron. Dark vs. dim silicon and near-threshold computing extended results. *Univ. of Virginia, Dept of Comp.Sci Technical Report*, 1, 2012.

978-1-5386-4881-0/18/$31.00 ©2018 IEEE