Adapt&Cap: A Framework for Unifying System and Application-level Adaptive Management

Can Hankendi
ECE Department
Boston University, Boston, MA
hankendi@bu.edu

Henry Hoffmann
Department of Computer Science
University of Chicago, Chicago, IL
hankhoffmann@cs.uchicago.edu

Ayse K. Coskun
ECE Department
Boston University, Boston, MA
acoskun@bu.edu

***Abstract*—Designing autonomous and adaptive techniques to manage data center resources has become inevitable to achieve sustainability and continue to meet increasing computational demand. However, uncoordinated use of adaptive techniques might lead to inefficient management algorithms. In this work, we propose a unified framework that takes advantage of both system and application-level adaptations to (1) improve performance under power caps, and (2) reduce power consumption under performance constraints. We implement Adapt&Cap on real servers and demonstrate up to 27% power reduction and 2.7x performance improvement compared to system or application-level only adaptation.**

I. Introduction

Infrastructural costs (e.g., power delivery, cooling capacity, electricity cost) and available hardware resources (e.g., CPU, disk size) determines the maximum achievable performance of a computing cluster. Optimizing the performance under such constraints (i.e., power, computation capacity) is critically important to improve energy efficiency and reduce computing costs.In data centers, constraining the peak power consumption of the servers via power capping is becoming a common practice for managing the energy costs and to comply with the power delivery limitations [1]. In order to reduce the administration and management costs, designing adaptive solutions has become necessary. Adaptive solutions not only reduce the costs under dynamic workload behavior, but also enable meeting the performance and power goals on the heterogeneous data center resources, which consist of various types of operation platforms and architectures. Traditional adaptive solutions employ system-level management knobs to comply with the power and performance requirements [2]. These system-level adaptive solutions use control knobs such as voltage/frequency selection (i.e., DVFS) or turning on/off cores. However, system-level solutions lack the ability to optimize the performance of the application running on the system depending on the architectural characteristics of the underlying platform. Adaptive applications address the performance optimization problem by dynamically configuring application parameters depending on the hardware properties and the performance goals [3]. As application and system-level decisions impact both the performance and the power consumption, uncoordinated decisions at these two levels can significantly hurt the overall energy efficiency of the system, which we discuss in detail in the following section.

In this work, we propose a unified framework that takes advantage of both system and application-level adaptability to (1) improve performance under power caps, and (2) reduce power consumption under performance constraints. Our specific contributions in this paper are as follows:

We first demonstrate how to improve the power/performance trade-off space by combining system and application-level adaptation. We propose a unified framework, Adapt&Cap, which combines system and application-level adaptations to improve performance while reducing the power consumption. We implement Adapt&Cap on real servers and demonstrate up to 27% power reduction and 2.7x performance improvement compared to system or application-level only adaptation. Next we provide our analysis on adaptive applications and systems. In Section III, we present the main components of Adapt&Cap and provide the details of our experimental setup. We preset our results collected on real servers in Section IV.

II. Motivation

On a cloud environment where resources are limited, power, performance and accuracy constraints are expected to be dynamically changing due to changing user requirements, energy pricing and cost management policies. Adaptive applications can meet these dynamically changing performance or accuracy targets by modifying a set of selected application parameters at runtime [4]. An adaptive application iteratively modifies its parameters until the used-defined constraints are met, while monitoring the performance and the accuracy impact to guide its decisions.

In Figure 1, we show the power and performance (i.e., seconds elapsed processing each frame) for x264 from the PARSEC suite [5] for three cases: (1) where we use adaptive capabilities only at the application-level (Application-level Only), (2) only at the system-level (System-level Only), and (3) at both application and system level (Coordinated). As Figure 1 shows, application-level decisions have minimal impact on the power, while providing the ability to adjust the performance for a wide range of targets. On the other hand, system-level decisions have a significant impact on power consumption, while providing a narrower performance range with respect to the adaptive application. Unifying the system and application level adaptability provides the Pareto-optimal curve for the power and performance space. These observations motivate the design of a unified framework that will minimize the power while meeting the performance constraints and maximize the performance while meeting the power constraints.

III. Adaptive Power Capping

In this section, we present the details of the proposed adaptive framework, Adapt&Cap. Adapt&Cap combines an

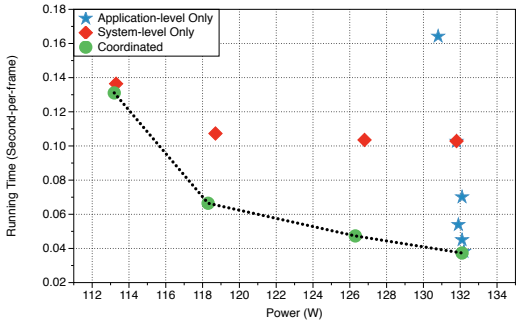

Fig. 1. Power and performance tradeoff space for various adaptive techniques on Intel Xeon E5 multicore server when running x264.

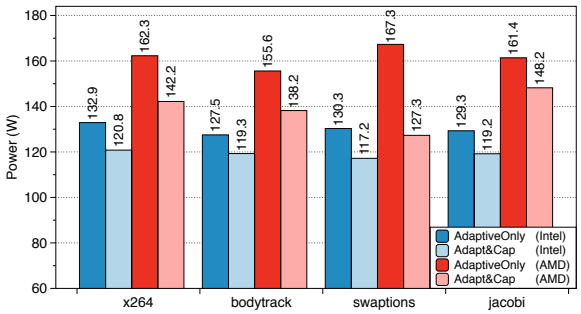

Fig. 2. Comparison of power consumption for Adapt&Cap and only adaptive application under dynamically changing performance constraints. Adapt&Cap reduces the power consumption up to 27% compared to only application-level adaptation.

application-level adaptive framework (i.e., Heartbeats) with a system-level adaptive power management framework to (1) maximize the performance under power constraints and (2) minimize the power consumption under performance constraints. Adapt&Cap maximizes performance through utilizing adaptive applications and minimizes the power consumption by employing system-level management. Adapt&Cap is built on top of the vCap framework [6] and extends the capabilities of vCap by taking advantage of the performance optimization capabilities of the adaptive-applications. Both power consumption and the heartbeat rates are periodically fed to the closed-loop controller to adjust and tune its decisions.

As a first step, Adapt&Cap discovers the adaptive states of the application within the code and chooses the state that achieves the highest performance. It then measures the performance and power consumption at the highest state (i.e., n). After deriving the power/performance relationship of an application at its best performing configuration, Adapt&Cap individually checks the power and performance constraints to adjust the CPU usage limits (CPU_{limit}) and to make thread packing decisions. For a given power cap, Adapt&Cap first computes the maximum achievable performance (HB_{cap}), then it computes the maximum amount of CPU resources that will not violate the power constraints (CPU_{limit}). Based on the CPU_{limit}, we derive the minimum number of active cores that can provide enough CPU resources to meet the computed CPUlimit.

IV. Experimental Results

In this section, we present the benefits of the Adapt&Cap framework on real-life servers. We test our framework under two scenarios that are (1) dynamically changing performance constraints and (2) dynamically changing power caps.We first test Adapt&Cap under dynamically changing performance constraints. We compare the benefits of Adapt&Cap with the adaptive versions of the applications that can track the performance requirements with its internal control through parameter adjustments (i.e., AdaptiveOnly). We only evaluate the parallel portions (regions of interest) of the PARSEC benchmarks (i.e., x264, bodytrack, swaptions) and the whole execution of jacobi.

In Figure 2, we report the average system-level power consumption of two real servers. Adapt&Cap significantly reduces the power consumption by utilizing system-level control knobs. Although adaptive capabilities of the applications are useful to meet the performance requirements, AdaptiveOnly consumes more or less the same amount of power regardless of the performance targets. On average, Adapt&Cap achieves up to 27% power reduction when compared to AdaptiveOnly approach.

In the second set of experiments, we evaluate Adapt&Cap under dynamically changing power caps and compare the performance of Adapt&Cap with vCap, which is an adaptive yet application agnostic power management technique and runs the default versions of the applications. For each system (i.e., Intel, AMD), we create separate power cap traces, as the power ranges of these two systems vary significantly. Adapt&Cap consistently outperforms the vCap and provides 1.72x performance improvements on average.

V. Conclusions

With the increasing degree of heterogeneity in todays data centers, it has become essential to design adaptive systems as well as adaptive applications that can optimize power and performance under various hardware configurations and/or dynamically changing constraints. In this work, we propose Adapt&Cap, which combines application and system-level adaptation to improve the energy efficiency. We implement Adapt&Cap on two real multi-core servers and show that unifying system and application-level adaptability improves the performance by 1.72x and reduces the power by 21% on average, when compared to system-only or application-only adaptations.

References

[1] R. Nathuji and K. Schwan, "VPM Tokens: Virtual Machine-aware Power Budgeting in Datacenters," in *International symposium on High Performance Distributed Computing (HPDC)*, 2008, pp. 119–128.

[2] K. Ma and X. Wang, "PGCapping: Exploiting Power Gating for Power Capping and Core Lifetime Balancing in CMPs," in *PACT*, 2012, pp. 13–22.

[3] H. Hoffmann, S. Sidiroglou, M. Carbin, S. Misailovic, A. Agarwal, and M. Rinard, "Dynamic Knobs for Responsive Power-aware Computing," in *ASLPOS*, 2011, pp. 199–212.

[4] P. Bailey, D. Lowenthal, V. Ravi, B. Rountree, M. Schulz, and B. De Supinski, "Adaptive configuration selection for power-constrained heterogeneous systems," in *Parallel Processing (ICPP), 2014 43rd International Conference on*, 2014, pp. 371–380.

[5] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The PARSEC Benchmark Suite: Characterization and Architectural Implications," in *International Conference on Parallel Architectures and Compilation Techniques (PACT)*, October 2008.

[6] C. Hankendi, S. Reda, and A. K. Coskun, "vCap: Adaptive Power Capping for Virtualized Servers," in *Proceedings of the 2013 International Symposium on Low Power Electronics and Design*, ser. ISLPED '13, 2013, pp. 415–420.