Exploiting Platform Heterogeneity for Power Efficient Data Centers

Ripal Nathuji
Georgia Institute of Technology
Atlanta, GA 30032
rnathuji@ece.gatech.edu

Canturk Isci
Princeton University
Princeton, NJ 08544
canturk@princeton.edu

Eugene Gorbatov
Intel Corporation
Hillsboro, OR 97124
eugene.gorbatov@intel.com

Abstract

It has recently become clear that power management is of critical importance in modern enterprise computing environments. The traditional drive for higher performance has influenced trends towards consolidation and higher densities, artifacts enabled by virtualization and new small form factor server blades. The resulting effect has been increased power and cooling requirements in data centers which elevate ownership costs and put more pressure on rack and enclosure densities. To address these issues, in this paper, we enable power-efficient management of enterprise workloads by exploiting a fundamental characteristic of data centers: "platform heterogeneity". This heterogeneity stems from the architectural and management-capability variations of the underlying platforms. We define an intelligent workload allocation method that leverages heterogeneity characteristics and efficiently maps workloads to the best fitting platforms, significantly improving the power efficiency of the whole data center. We perform this allocation by employing a novel analytical prediction layer that accurately predicts workload power/performance across different platform architectures and power management capabilities. This prediction infrastructure relies upon platform and workload descriptors that we define as part of our work. Our allocation scheme achieves on average 20% improvements in power efficiency for representative heterogeneous data center configurations, highlighting the significant potential of heterogeneity-aware management.

1 Introduction

Power management has become a critical component of modern computing systems, pervading both mobile and enterprise environments. In data centers, power consumption has become a significant issue, stimulating a variety of research for server systems [2]. Increased performance requirements in data centers have resulted in elevated densities enabled via consolidation and reduced server form factors. This has in turn created challenges in provisioning the necessary power and cooling capacities. For example, current data centers allocate nearly 60 Amps per rack, a limit that is likely to become prohibitive for future high density rack configurations such as blade servers, even if the accompanying cooling issues can be solved [19]. In addition, the financial overheads of utilizing these resources are another impetus for incorporating management capabilities. A 30,000 square feet data center with a power consumption of 10MW requires a cooling system which costs $2-$5 million [17]. In such a system, the cost of running the air conditioning equipment alone can reach $4-$8 million a year [19]. Coupled with the elevated electricity costs from increasingly high performance servers, these effects can substantially affect the operating costs of a data center.

The trends in power/cooling delivery and cost highlight the need for support in data centers for power and thermal management. Some of the previous work on server management has focused on managing heat generation during thermal events [17] or utilizing platform power management support, such as processor frequency scaling, for power budgeting [8, 19]. In this paper, we address an orthogonal question: Given that thermal and power constraints are managed, how can we allocate workloads to platforms intelligently to improve the power efficiency of a data center?

Typically, data centers statically allocate sets of platforms to applications based upon peak load characteristics to maintain isolation and to provide performance guarantees. With the continuing growth in capabilities of virtualization solutions such as Xen [1], the necessity of such offline provisioning is removed. Indeed, by allowing for flexible and dynamic migration of workloads across physical resources [6], the use of virtualization in future data centers enables a new avenue of management and optimization. Our approach begins to leverage some of these capabilities to enhance power efficiency by taking advantage of the ability to assign virtualized applications to varying sets of underlying hardware platforms.

Throughout their lifetimes, data centers continually upgrade servers due to failures, capacity increases, and migrations to new form factors [11]. Over time, this leads to data

centers comprised of a range of heterogeneous platforms with different technologies, power, performance and thermal characteristics, and power management capabilities. When assigning platforms to application workloads in these heterogeneous environments, power efficiency can vary significantly based on the particular allocation. For example, by assigning a memory bound workload to a platform that performs dynamic voltage and frequency scaling (DVFS), run-time power consumption can be reduced with minimal impact to performance [16]. To obtain this power-friendly behavior in data centers, we develop a heterogeneity-aware workload allocation architecture.

Intelligent mapping of applications to underlying platforms is dependent upon the availability of relevant information about workloads and hardware resources. In our scheme, we extend the use of *workload* and *platform descriptors* for this purpose, which are then used by a *predictor* component that estimates the achievable performance and power savings across the different platforms in the data center. These predictions are finally used by an *allocation layer* to map workloads to a specific type of platform. This overall infrastructure is evaluated using data center configurations consisting of variations upon four distinct platforms. The main contributions of our work are: (i) Considering the use of platform heterogeneity including differences in power management support for improved power efficiency, (ii) Design of an allocation infrastructure which relies upon workload and platform descriptors to perform informed mappings of hardware to virtualized workloads, (iii) Evaluation of our system on state-of-the art platforms including Intel® Core™microarchitecture based hardware. Our results show average improvements of 20% in data center power efficiency. These results highlight the efficacy of our approach, and demonstrate the benefits of exploiting platform heterogeneity for improved power efficiency.

The rest of the paper is organized as follows: Section 2 reviews related work and how our work fits into the larger landscape of research. We next discuss the opportunities with heterogeneity-aware workload allocation in Section 3, followed by an overview of our allocation architecture and descriptor design in Section 4. After describing our hardware and application assumptions in Section 5, we delve into the details of the prediction and allocation policies in Sections 6 and 7 respectively. Section 8 presents our evaluation results, followed by concluding remarks and discussions of future work in Section 9.

2 Related Work

A variety of mechanisms have been developed which provide power and thermal management support within a single platform. Brooks and Martonosi proposed mechanisms for the enforcement of thermal thresholds on the processor [3]. Processor frequency and voltage scaling based upon memory access behavior has been shown to successfully provide power savings with minimal impact to applications. Resulting solutions include hardware based approaches [16] and OS-level techniques, which set processor modes based on predicted application behavior [13]. Power budgeting of SMP systems with a performance loss minimization objective has also been implemented via CPU throttling [14]. Though these types of approaches allow for local management of nodes, they don't address the issues which arise when considering multiple systems.

At the data center level, incorporating temperature-awareness into workload placement has been proposed by Moore *et al.* [17], along with emulation environments for studies of thermal implications of power management [10]. Chase *et al.* discuss how to reduce power consumption in data centers by turning servers on and off based on demand [4]. Utilizing this type of cluster reconfiguration in conjunction with DVFS [7] and the use of spare servers [18] has been investigated as well. Enforcing power budgets within data centers by allocating power in a non-uniform manner across nodes has been shown to be an effective management technique [8]. Techniques for enforcing power budgets at blade enclosure granularities have also been discussed [19]. The approach presented in our work can be used in conjunction with these methods to improve power efficiency in data centers.

Heterogeneity has been considered to some degree in prior work, including the evaluation of heterogeneous multi-core architectures [15]. In cluster environments, a scheduling approach for power control has also been proposed for processors with varying fixed frequencies and voltages [9]. A power efficient web server with intelligent request distribution in heterogeneous clusters is another example which considers leveraging heterogeneity in enterprise systems [11]. Our vision goes beyond these various methods by considering not just differences amongst performance capabilities of platforms, but also in the power management capabilities they may support.

3 Exploiting Heterogeneity for Increased Power Efficiency

Data center deployments are inherently heterogeneous. Upgrade cycles and replacement of failed components and systems contribute to this heterogeneity. In addition, new processor and memory architectures appear every few years and reliability requirements are becoming ever more stringent. These trends have driven update cycles in large data centers to less than two years. A recent survey of data center managers shows that 90% of the facilities are expected to upgrade their compute and storage infrastructure in the next two years. Figure 1 shows a distribution of different systems in a representative enterprise data center. As the figure shows, the data center contains nine different gen-

erations of systems that have either (1) different processor architectures, cores and frequencies, (2) varying memory capacity and interconnect speeds, or (3) different I/O capabilities. While all systems support the same software stack they have very different and often asymmetric performance and power characteristics.

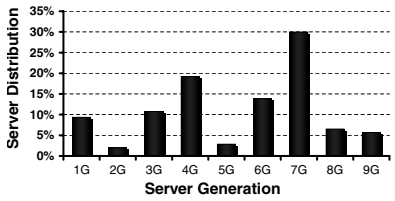


Figure 1. Data center heterogeneity.

Traditionally, the non-uniformity of systems in a data center has been characterized by different levels of performance and power consumption. However, recently, another dimension has been added to this heterogeneity. Current server platforms are beginning to support various thermal and power management capabilities. Processors support DVFS and aggressive sleep states to conserve CPU power. New memory power management implementations allow different DRAM devices to go to lower power states when inactive, and enable bandwidth throttling for thermal protection. Finally, server power supplies exhibit different conversion efficiencies under different loads directly impacting the overall power efficiency of the system. Since power efficiency has become one of the main thrusts in enterprise systems, we expect component and platform vendors to continue introducing new power and thermal management capabilities into their products, including I/O and system buses, chipsets, and network and disk interfaces, making future platforms even more heterogeneous.

Previous work has proposed different approaches for energy-efficient workload allocation in clusters in data centers, but none have accounted for system level power management and thermal characteristics. Therefore, the workload allocations proposed by previous approaches will yield less than ideal results since they are completely unaware of power and thermal management effects on system performance and power consumption. To illustrate this phenomenon, we experimentally compare two dual processor systems, A and B, running two different workloads as shown in Table 1. The differences between the two systems are in the power supply unit (PSU) and processor power management capabilities. System A has a less efficient power supply at light load and has processors with limited power management support. System B, on the other hand, has a high efficiency power supply across all loads and processors that support a rich set of power management capabilities.

We measure power consumption on these platforms using two different synthetic workloads: one with full utilization ($W1$) and one with a very low level of utilization ($W2$)

	System A		**System B**	
	W1	W2	W1	W2
CPU Power	90W	40W	90W	20W
System Power	160W	120W	160W	120W
PSU Efficiency	86%	70%	87%	80%
Total Power	**291W**	**229W**	**287W**	**175W**

Table 1. Power consumption of System A and System B with workloads W1 and W2.

on both systems. $W1$ consumes about the same amount of power on both platforms. However, allocating the low-utilization $W2$ to system A leads to very power inefficient execution. Since A does not support power management and has low PSU efficiency at light load, its total system power is more than 50W higher than system B. While both systems meet the performance demand of both workloads, power-aware resource allocation can decrease total power by more than 10%. For a large data center, 10% overall power reduction translates into millions of dollars in savings in utility costs. As this example shows, a full knowledge of system power and supported power management features is required to efficiently allocate workloads across such heterogeneous systems. As we describe in the following sections, our allocation architecture is designed to address these needs.

4 Scalable Enterprise and Data Center Management

Our previous discussions have motivated the need to augment the behavior of data centers to improve power efficiency by leveraging the heterogeneity of platform capabilities. We extend this support by developing a heterogeneity-aware workload allocation infrastructure which utilizes the flexibility of rapidly developing virtualization technologies. Virtualization provides capabilities and abstractions that significantly impact the landscape of enterprise management. For example, by providing performance isolation, it is possible to run multiple virtual machines (VMs) within a given physical platform without interference among applications. Moreover, by encapsulating application state within well defined virtual machines, migration of workloads among resources can be performed easily and efficiently. A more powerful contribution of virtualization, though, is the ability to combine resources across physical boundaries to create virtual platforms for applications, providing a *scalable enterprise* environment. We assume the existence of this flexible and powerful virtualization support in designing our management system.

In the future, data centers will be service-oriented where applications and workloads may be submitted dynamically by subscribers/clients. The types of applications in this scenario require management actions to be performed at coarse time granularities, where allocation management may be

performed as rarely as on a daily basis. One can imagine how such a data center might be managed with the typically used assignment approaches. Each day the pool of applications and service level agreements (SLAs) which specify their required performance, in metrics such as throughput or response time, are compiled. Applications are then assigned to platforms using a simple load balancing scheme based upon utilization or queue lengths, possibly even accounting for differences in the performance of the systems [20], so that SLAs are met. This approach clearly leaves room for improvement since it does not consider power in any way, but instead focuses on obtaining resources to meet application performance needs. Our approach addresses this weakness by performing heterogeneity aware allocations.

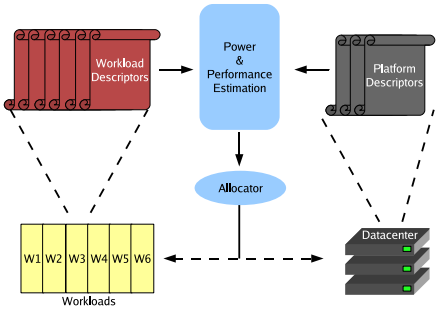


Figure 2. Heterogeneity-aware workload allocator architecture.

The architecture of our heterogeneity-aware allocation system can be organized into three major components: (1) platform/workload descriptors, (2) a power/performance predictor, and (3) an allocator, as shown in Figure 2. We use platform and workload descriptors to provide our workload allocator with the differences amongst workloads and platforms. These descriptor inputs are utilized by the predictor to determine: (i) the relative performance of workloads on different types of platforms (ii) the power savings achievable from platform power management mechanisms. Coupled with coarse platform power consumption information (obtained via online power monitoring), our third component, the allocator, performs the assignments of workloads to the available resources.

The purpose of platform descriptors is to convey information regarding the hardware and power management capabilities of a machine. A platform descriptor is made up of individual modules, representing system components, as shown in Figure 3(a). Each module specifies the type of component the module refers to, such as processor, memory subsystem, or power supply. Within each of these modules, then, various component parameters are defined. For example, a module describing the processor component may have attributes such as microarchitecture family, frequency, and available management support.

Workload descriptors are also structured in modules,

```
Component: Processor
  Architecture:     Netburst
  No_Cores:         4
  Frequency:        3.2 GHz
  p-state_support: Yes

Component: Memory
  Cache_size:  2MB
  Memory_type: DDR2
  Memory_size: 8GB
.
.
.
```

(a) Platform Descriptor

```
Attribute: MPI
  val1: <Memory.Cache_size=2MB>
  val2: <Memory.Cache_size=4MB>

Attribute: CPI_CORE
  val1: <Processor.Architecture=
         NetBurst>
  val1: <Processor.Architecture=
         Core Microarchitecture>
.
.
.
```

(b) Workload Descriptor

Figure 3. Descriptor examples.

headed with attribute declarations. Within each module, a list of values for that attribute is provided. As workload attributes often vary with the platform on which they execute, our descriptor design allows multiple attribute definitions, where each definition is predicated with component parameter values that correlate back to platform descriptors. Figure 3(b) illustrates the structure of the resulting workload descriptor. We explain the meaning of the MPI and CPI_{CORE} attributes in subsequent sections.

In our infrastructure, the descriptor information is provided in a variety of ways. Platform descriptor information can be made readily available using platform support such as ACPI [12], and possibly also with some administrative input. To provide the required workload descriptors, we profile workloads on a minimal set of *orthogonal platforms*, with mutually exclusive component types. We then use an analytical prediction approach to project workload characteristics on all available platforms. As we discuss in detail in Section 6, this approach provides accurate predictions that scale with increased amounts of heterogeneity.

5 Methodology

5.1 Platform Hardware

Our hardware setup consists of four types of rack mounted server platforms summarized in Table 2, where LLC denotes last-level cache size. All four types of platforms contain standard components and typical configurations that entered production cycles in the last eighteen months. The platform names are based on their processor code name in this paper. All four platforms are dual-processor systems. Woodcrest, Sossaman, and Dempsey are CMP dual-core processors and Irwindale is a 2-way SMT processor supporting Hyper-Threading Technology. All platforms have 8GB of memory. Woodcrest and Dempsey support Fully Buffered DIMM (FBD) memory with a 533MHz DDR2 bus while Sossaman and Irwindale support unregistered DDR2 400MHz memory. Woodcrest and Dempsey have independent FSB architectures with two branches to memory and two channels per branch.

All four types of systems are heterogeneous in a sense that each has a unique combination of processor architecture and memory subsystem. If we assume that Intel Core microarchitecture/Pentium® M and NetBurst constitute two types of processors and LLC-4MB/FSB-

	Woodcrest	Sossaman	Dempsey	Irwindale
Processor	3GHz/4MB LLC Core architecture	2GHz/2MB LLC Pentium® M	3.7GHz/4MB LLC NetBurst/P4	3.8GHz/2MB LLC NetBurst/P4
FSB	1067 MHz Dual FSB	800 MHz	1067 MHz Dual FSB	800 MHz
Chipset	Blackford	Lindenhurst	Blackford	Lindenhurst
Memory	DDR2-533 FBD	DDR2-400	DDR2-533 FBD	DDR2-400

Table 2. Platform characteristics.

1066/FBD-533 and LLC-2MB/FSB-800/DDR2-400 constitute two types of memory, all four platforms can be mapped as having unique processor/memory architecture combinations. This results in a four quadrant heterogeneity space as shown in Figure 4. Note that all four platforms also have vastly different power and performance characteristics. For example Intel Core microarchitecture is superior to NetBurst both in terms of performance and power efficiency. FBD based memory, on the other hand, provides higher throughput in our systems at the expense of elevated power consumption due to increased DDR2 bus speed and the power requirements of the Advanced Memory Buffer (AMB) on the buffered DIMMs.

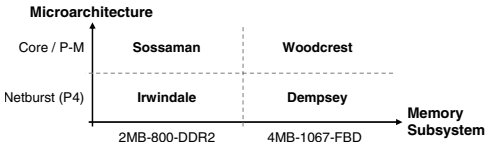


Figure 4. Heterogeneity quadrants.

The four platforms described occupy four quadrants of a heterogeneity space with dimensions of microarchitecture heterogeneity and memory subsystem heterogeneity. We refer to this initial level of heterogeneity as *"across-platform heterogeneity"*. However, in addition to this, all these server platforms also support chip-level DVFS. This leads to a second degree of heterogeneity, where one type of platform, can have instances in a data center that are configured to operate at different frequencies. We refer to this as *"within-platform heterogeneity"*. As process variations increasingly result in the *binning* of produced chips into different operating points, this within-platform heterogeneity becomes an inherent property of the general data center landscape. Finally, many of these platforms may incorporate some processor dynamic power management (DPM) techniques that adaptively alter platform behavior at runtime. This creates a third source of heterogeneity, *"DPM-capability heterogeneity"*, where platforms with built-in DPM hooks exhibit different power/performance characteristics from the ones with no DPM capabilities. In Table 3, we show how these three levels of heterogeneity quickly escalate the number of distinct platform configurations in a data center scenario.

Our power measurements have been performed using the Extech 380801 power analyzer. The power was measured at the wall and represents total AC power consumption of the entire system. The power numbers presented in this paper are obtained by averaging the instantaneous system power consumption over the entire run of each workload. Our assumption is that infrastructure support for monitoring power consumption will be utilized to obtain this type of workload specific power characteristics online, instead of parameterized models. For example, all power supplies, which adhere to the latest power supply monitoring interface (PSMI) specification support out-of-band current/voltage sampling allowing for per platform A/C power monitoring reflected by our actual power measurements.

	Across-Platforms				Within-Platform	DPM-Capability		Heterogeneous Configuration
	Microarchitecture		Memory		Frequency [GHz]	Enabled	Disabled	
	Core	Netburst	FBDIMM	DDR-2				
Woodcrest	X		X		3.0	x		1
							x	2
					2.6	x		3
							x	4
					2.3	x		5
							x	6
					2.0	x		7
							x	8
Sossaman	X			X	2.0	x		9
							x	10
					1.6	x		11
							x	12
					1.3	x		13
							x	14
					1.0	x		15
							x	16
Dempsey		X	X		3.7	x		17
							x	18
					3.2	x		19
							x	20
Irwindale		X		X	3.8	x		21
							x	22
					3.2	x		23
							x	24
					2.8	x		25
							x	26

Table 3. Levels of heterogeneity in our experimental platforms.

5.2 Application Model

When managing power efficiency in computing environments, improvements can be attained with a variety of approaches. In this work, we concentrate on improving power consumption, while the baseline application performance is maintained. In other words, we maximize the performance per watt, while holding performance constant. We consider application performance in terms of throughput, or the rate at which transaction operations are performed. Therefore it is not the execution time of each transaction that defines performance, but the rate at which multiple transactions can be sustained. This type of model is representative of applications such as web servers or payroll systems.

The goal of our allocator is to evaluate the power-efficiency tradeoffs of assigning a workload to a variety of platforms. Since the performance capabilities of each platform are different, the execution time to perform a single operable unit, or atomic transaction, varies across them. As previously mentioned, virtualization technologies will allow us to extend the physical resources dedicated to applications when necessary to maintain performance by increasing the number of platforms used to execute transactions. In particular, transactions can be distributed amongst nodes until the desired throughput is reached.

For our analysis, we consider applications which mimic the high performance computational applications common to data center environments. There are two aspects of these workloads that are captured in our experimental analysis. First, these workloads are inherently transactional, such as

the previous financial payroll example or the processing of risk analysis models across different inputs common to investment banking. Second, with the ability to incorporate large amounts of memory into platforms at relatively low costs, these applications often execute mostly from memory, with little to no I/O. To obtain both of these characteristics, while also providing deterministic and repeatable behavior for our experimentation, we utilize benchmarks from the SPEC CPU2000 suite as representative examples of transaction instances. SPEC benchmarks allow for the isolation of processor and memory components, while also generating different memory loads. Indeed, many SPEC benchmarks exhibit significant measured memory bandwidth of 5-8 GB/sec on our systems. In order to provide an unbiased workload set, we include all SPEC benchmarks in our experiments. For each application, we specify an SLA in terms of required transaction processing rate, equal to the throughput achievable on the Woodcrest platform.

6 Workload Behavior Estimation

The power/performance predictor component of our heterogeneity-aware workload allocation framework can be implemented in different fashions. For example, one can profile a set of microbenchmarks on all platform configurations and develop statistical mapping functions across these configurations. However, as the platform types and heterogeneity increase, the overhead of such approaches can be prohibitive. Instead, we develop a predictor that relies on the architectural platform properties and adjusts its predictions based on the heterogeneity specifications. We refer to this model as the *"Blocking Factor (BF) Model"*.

6.1 Blocking Factor Model

The BF model simply decomposes execution cycles into *CPU cycles* and *memory cycles*. CPU cycles represent the execution with a perfect last-level cache (LLC), while memory cycles capture the finite cache effects. This model is similar to the *"overlap model"* described by Chou *et al.* [5]. With the BF model, the CPI of a workload can be represented as in Equation 1. Here CPI_{CORE} represents the CPI with a perfect LLC. This term is independent from the underlying memory subsystem. CPI_{MEM} accounts for the additional cycles spent for memory accesses with a finite-sized cache.

$$CPI = CPI_{CORE} + CPI_{MEM} \quad (1)$$

The CPI_{MEM} term can be expanded into the architecture and workload specific characteristics. Based on this, the CPI of a platform at a specific frequency f_1 can be expressed as in Equation 2. Here MPI is the memory accesses per instruction. This is dependent on the workload and the LLC size. L is the average memory latency, which depends on the memory subsystem specifications and BF is the *blocking factor* that accounts for the overlapping concurrent execution during memory accesses, which is a characteristic of the workload.

$$CPI(f_1) = CPI_{CORE}(f_1) + MPI \cdot L(f_1) \cdot BF(f_1) \quad (2)$$

To estimate how the CPI of an application changes with frequency, we need to consider how the independent parameters vary. CPI_{CORE} is independent of frequency, as cycles spent with perfect LLC do not change with frequency. MPI is a feature of the workload and does not change significantly with frequency. The actual memory latency is constant in time and does not scale with CPU frequency. Therefore, the cycle memory latency should scale with frequency. Finally, for the BF parameter we experimentally compared prediction accuracies using both a constant BF and one that varies with frequency. Both approaches perform comparably in terms of CPI prediction accuracy (with 1.2% average prediction errors). Therefore, to simplify our workload descriptors we assume BF is constant across frequencies. Based on these observations, the CPI of a platform at a different frequency f_2 can be expressed as in Equation 3.

$$CPI(f_2) = CPI_{CORE}(f_1) + MPI \cdot L(f_1) \cdot (f_2/f_1) \cdot BF(f_1) \quad (3)$$

With this interpretation of the BF model, by simply knowing the CPI_{CORE}, MPI and BF of a workload at a specific frequency, we can predict its behavior on all other instances of within-platform heterogeneity for a platform. However, the more interesting application of the BF model is for the across-platform heterogeneity. Here the natural decoupling of the microarchitectural and memory subsystem differences in the BF model enables us to estimate application performance on a platform lying on a different corner of the memory and microarchitecture heterogeneity space. Among our four experimental platforms, two of these can be chosen as *"orthogonal platforms"*, which span the two opposite corners of the across-platform heterogeneity. For our experiments, Sossaman and Dempsey platforms satisfy this condition, as they have mutually exclusive microarchitectural and memory properties. Then, the characteristics of the remaining two platforms can be composed from the subcomponents of the two orthogonal platforms. For example a Woodcrest platform can be approximated as the composition of the microarchitectural features of the Sossaman platform and the memory subsystem of the Dempsey platform. Conversely, the Irwindale platform can be considered as a composition of Sossaman-like memory and Dempsey-like microarchitecture properties. Note that although LLC features are architectural features, these are considered as part of the memory subsystem as their effect pertains to the memory CPI. With this division of platforms into *"orthogonal"* and *"derived"* platforms, we can simply gather workload characteristics on single instances of the orthogonal platforms, and project application behavior on all other platform configurations in the data center.

Considering across- and within-platform heterogeneity,

by determining the CPI_{CORE}, MPI and BF for a workload on two specific frequency settings of Sossaman and Dempsey platforms, we can predict the workload behavior on all—total of 13—configurations of all the platforms. To determine the workload behavior on another instance of the orthogonal platforms, we simply use the within-platform prediction method described above. To predict the behavior on a derived platform, we use the corresponding CPI_{CORE} and CPI_{MEM} components from the orthogonal platforms with the memory latency L of the derived platform. For example, Equation 4 shows how the CPI for a Woodcrest platform at frequency f_1 can be predicted from Sossaman and Dempsey descriptors.

$$CPI(W@f_1) = CPI_{CORE}(S) + MPI(D) \cdot L(W@f_1) \cdot BF(D) \quad (4)$$

Here, $CPI(W@f_1)$ is the CPI of Woodcrest at frequency f_1, $CPI_{CORE}(S)$ is the perfect LLC CPI of Sossaman, $MPI(D)$ is the memory accesses per instruction for Dempsey, $L(W@f_1)$ is the memory latency of Woodcrest at frequency f_1, and $BF(D)$ is the blocking factor observed from Dempsey. With this approach, we can provide reasonably accurate predictions of workload behavior on different platforms, without actually accessing the derived platforms.

The final heterogeneity type that our predictor must support is the DPM-capability heterogeneity. For this, we consider a platform which enables DVFS during memory bound execution regions of an application. We implement this functionality as part of OS power management based on prior work [13]. We modify the operating system so that it utilizes performance counters to detect and predict memory bound phases. Phases are further distinguished based upon degree of memory boundedness and are associated with a processor frequency to execute at. We tune the DPM enabled system so that there is negligible impact on application performance.

To incorporate DPM awareness, we extend the predictor component to estimate the potential power savings that can be attained when executing a workload on a DPM enabled platform. Our experiments show that there is a strong correlation between the MPI of a workload and its power saving potential. Therefore, we utilize the MPI attribute in the workload descriptors to predict the power saving potentials of workloads on DPM enabled platforms.

As we have previously described, the fundamental vision of our energy-efficient allocation framework is to decide upon workload-platform assignments by utilizing well-defined workload and platform descriptors. The BF model fits very well in this vision. The independent parameters that we require on the orthogonal platforms for each workload, CPI_{CORE}, MPI and BF constitute the workload descriptors. On the other hand, specific characteristics of each platform, memory latency (L), memory subsystem specifications including LLC, microarchitectural features, frequency states and DPM capabilities constitute the platform descriptors. During each new encounter of a workload, the predictor uses these descriptors to extract the relevant CPI components and predicts the workload and DPM power saving behavior on all the underlying platforms.

6.2 Prediction Results

Here we present the achieved prediction accuracies with our experimental platforms. In our evaluations, we first compare the predicted workload performance to the actual workload performance acquired by performance counters. Figure 5 shows the actual and predicted execution times across four different experimental platforms. Here, the top two plots show the results for the orthogonal platforms, Sossaman and Dempsey respectively. The lower two plots show the derived platforms, Woodcrest and Irwindale respectively. Although we have performed our evaluations at all available frequency settings for each platform, in this figure we only show one set of results per platform for brevity. The observed prediction accuracies are consistent in all other operating frequencies.

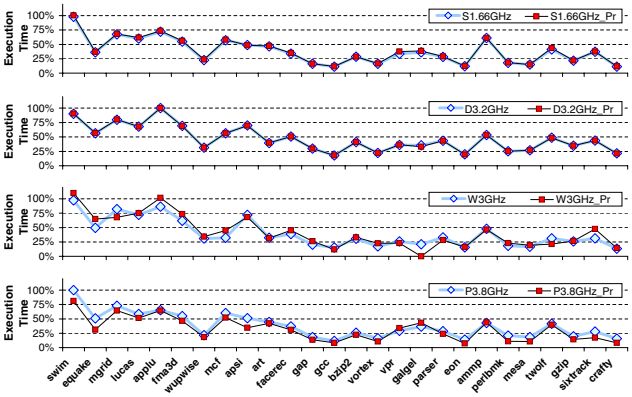


Figure 5. Predicted and measured execution times for the experimented platforms (normalized to maximum measured execution time on the platform). The plots show Sossaman, Dempsey, Woodcrest and Irwindale platforms from top to bottom.

For the orthogonal platforms, the BF model uses the within-platform estimations to derive the execution times at all frequencies. In these platforms, the BF model can very accurately predict performance with an average prediction error of 2%. Interestingly, the predictions track actual execution times very well also in the derived platforms, even though the BF model does not rely on any actual measured application behavior on these platforms. By only using the independent parameters observed on Sossaman and Dempsey, the BF model can produce accurate projections of application behavior on Woodcrest and Irwindale, simply by leveraging the architectural and memory system similarities between the derived and orthogonal platforms. For these derived platforms, the average predic-

tion error is 20%. These results show that our prediction method with the BF model can produce reasonable approximations to workload behavior under within- and across-platform heterogeneity. Although the across-platform prediction errors are relatively higher, the figures show that they successfully preserve the performance relations across benchmarks, and more importantly across platforms. In the following sections, we show that this prediction methodology provides sufficient accuracy to represent workload behavior and leads us to achieve close to optimal allocations with our heterogeneity-aware allocator.

In terms of estimating possible DPM savings, Figure 6 shows that our *MPI* based prediction approach effectively captures the power saving potentials of different workloads and successfully differentiates applications that can benefit significantly from being allocated to a DPM enabled machine. As we describe in Section 7, we use this predictor in our allocation decisions to choose workloads that should be assigned to the DPM enabled platforms.

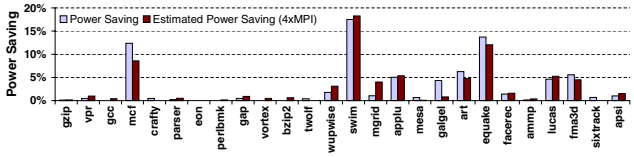


Figure 6. Measured and predicted power savings on a DPM-enabled platform.

7 Allocation Policies

After processing the workload and platform descriptors, and utilizing our BF model for performance prediction, the final step is to perform allocation of resources to a set of applications in a data center. We develop a greedy policy towards allocating workloads in our evaluation. In particular, for each application i, we associate a cost metric for executing on each platform type k. Workloads are then ordered based on their maximum cost metric across all platform types into a scheduling queue. The allocator then performs application assignment based on this queue, where applications with higher worst-case costs have priority. The platform type chosen for an application is a function of this cost metric across the available platforms as well as the estimated DPM benefits.

There are two workload specific values that may be relevant to allocation policy decisions. First, there is the number of platforms of type k required to execute a workload i, $N_{i,k}$. This value is clearly a function of the performance capabilities of the platform as well as the SLA requirement of the workload. In addition, there is the actual power cost of executing the workload on the platform type, $P_{i,k}$. This value depends upon both the number of required platforms, the power characteristics of the platform, as well as the DPM savings achievable when running the workload on the platform type.

Both $N_{i,k}$ and $P_{i,k}$ can be analytically defined provided the transaction based application model utilized in our work. For each application i, the service level agreement (SLA) specifies that X_i transactions should be performed every Y_i time units. If $t_{i,k}$ is the execution time of a transaction of application i on platform k, the resulting number of platforms required to achieve the SLA can be expressed with Equation 5.

$$N_{i,k} = \lceil \frac{X_i}{\lfloor \frac{Y_i}{t_{i,k}} \rfloor} \rceil \tag{5}$$

The $t_{i,k}$ values are provided from the performance predictor. It should be noted that there is a discretization in $N_{i,k}$ which is due to the fact that individual atomic transactions cannot be parallelized across multiple platforms.

Given $N_{i,k}$ and the platform power characteristics, the power cost of executing a workload can also be estimated. Here, we initially ignore any power savings from DPM mechanisms which may be employed on the underlying platforms. We can then estimate $P_{i,k}$ using the approximate active and idle power characteristics of a platform, P_{A_k} and P_{I_k} respectively, as in Equation 6.

$$P_{i,k} = \frac{1}{Y_i}(X_i \cdot t_{i,k} \cdot (P_{A_k} - P_{I_k}) + P_{I_k} \cdot N_{i,k} \cdot Y_i) \tag{6}$$

Both $P_{i,k}$ and $N_{i,k}$ are plausible candidates for a cost metric in our allocation policy. $P_{i,k}$, though, can be sensitive to errors in the prediction layer since variations in $t_{i,k}$ directly affect it. On the other hand, $N_{i,k}$ is better able to handle errors due to the inherent discretization performed. Therefore, we choose $N_{i,k}$ as the cost metric in our policy layer.

Given the use of $N_{i,k}$ as our cost metric, our allocation approach first determines the platform types of which (i) there are enough available systems to allocate the workload and (ii) the cost metric is minimized. We then use DPM savings to determine whether a more power efficient platform alternative should be used between those with the same cost value. In other words, if there are multiple platform types for which an application has the same $N_{i,k}$ value, we utilize a DPM specific threshold to decide whether or not it should be scheduled to a DPM enabled platform type. As we demonstrate in the following section, this threshold based approach can be effective in identifying workloads that can take advantage of DPM capabilities.

8 Evaluation

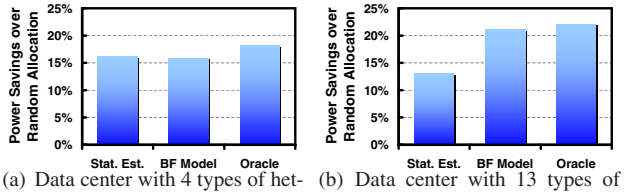
In order to evaluate our heterogeneity-aware allocation approach, we performed power and performance measurements of our SPEC based representative transactional workloads across each type of platform. To scale these results to the number of platforms present in data centers, this measured data was extrapolated analytically using a data center allocation simulator which combines real power and performance data, prediction output, and allocation policy def-

initions to calculate power efficiency in various data center configurations. In the simulator, we provide the output of the predictor as input to the allocation policy. We always assume that the platforms which are profiled are the 2GHz Sossaman platform and the 3.7GHz Dempsey systems. Since we assume the workload attributes are profiled accurately on these systems, for fairness we also assume that for these two platforms performance data is obtained via profiling as well and is therefore known perfectly. We then consider three different scenarios: (1) all other platform performance information is known perfectly (oracle) (2) our BF model is used to predict performance for the remainder of platforms as described in Section 6 (BF model) (3) incorporating a simple statistical regression approach (Stat. Est.). For this regression method we profile a subset of applications across all platforms to obtain linear performance prediction models parameterized by variables that can be obtained by profiling a workload on the 2GHz Sossaman and 3.7GHz Dempsey systems (CPI, MPI, etc.). The regression models can then be used to predict performance of any application. The baseline allocation scheme we compare against is a random one, since it closely estimates the common round-robin or utilization based approach.

The efficiency improvements achievable in a data center are also dependent upon the mix of applications that are in the system. To obtain our results, we randomly pick applications and allocate using the random approach until no more workloads can be scheduled. Using the resulting set of workloads, we then evaluate the power consumption when using our prediction and allocation policies, and compare against the random allocation result. This is repeated a hundred times for each of our data points.

We first look at the benefits we achieve with our heterogeneity-aware allocator in data center configurations with varying amounts of heterogeneity, and no DPM support. Considering just the four base platforms Woodcrest, Sossaman, Dempsey, and Irwindale running at their highest frequency settings, we create data center configurations with equal numbers of each. Trends were consistent across various data center sizes, so for space we include results with 1000 platforms of each type. The resulting normalized power consumption for the data center is shown in Figure 7(a). Since performance is maintained for all applications involved, reduced power consumption correlates directly to improved power efficiency. We see from the figure that with perfect workload information, power consumption is reduced by 18% when compared to the random allocation policy. We also see that our prediction-based workload allocations, with incomplete workload descriptors are able to achieve power savings close to the oracular allocator. The statistical model performs well with this limited amount of heterogeneity, but more importantly, our analytical BF model based prediction attains savings of 16%, highlighting its efficacy.

Next, we also include within-platform heterogeneity in our analysis with the frequency variations of our platforms. The resulting data center has 13 types of platforms, and power consumptions vary with allocation as shown in Figure 7(b). The first interesting observation we make is that increased heterogeneity allows us to achieve improved benefits over a simple random approach. Indeed, we see improvements of 22% with perfect knowledge and 21% using our BF based prediction. We also observe a significant difference between the statistical and analytical prediction schemes. The regression approach is unable to scale in terms of accuracy with increased heterogeneity, whereas the BF approach again achieves close to optimal power savings.



(a) Data center with 4 types of heterogeneous platforms

(b) Data center with 13 types of heterogeneous platforms

Figure 7. Data center power improvements.

In order to evaluate how well our allocator can exploit DPM support, we extend the thirteen platform type configuration with an additional Woodcrest 3GHz platform which provides DPM support. We again find that our BF prediction method can provide improved savings over the statistical approach as shown in Figure 8. To more closely determine our ability to exploit DPM mechanisms, we also evaluate the power consumption of the thousand DPM-enabled platforms (all of which are active). We find that our BF model based allocation is able to improve the power efficiency of these platforms by 3.3%. This illustrates the potential of our heterogeneity-aware allocator to provide additional benefits when platforms vary in the power management they support.

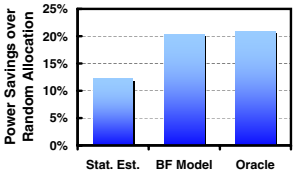


Figure 8. DPM-aware power savings.

9 Conclusions and Future Work

Power management in data center environments has become an important area of research. Power delivery and cooling limitations are quickly becoming a bottleneck in the provisioning of performance required by increasingly demanding applications. In this paper we address the problem of improving power efficiency when executing these workloads. In particular, we make use of the management flexibility afforded by virtualization solutions to exploit the

natural heterogeneity of platforms in data centers, including variances in dynamic power management support that may be available. We introduce a three phase approach to mapping workloads to underlying resources to improve power efficiency consisting of structured platform and workload descriptors, a prediction component to estimate the performance and power characteristics of various workload to platform mappings, and finally an allocator which utilizes policies and prediction results to perform decisions.

Our results underscore two major conclusions. First, we show that by intelligently considering the varying power management capabilities of platforms, the ability for these systems to obtain power savings using their management mechanisms can be vastly improved when compared to a simple round robin assignment model. Using representative data center configurations consisting of older P4 based platforms up to cutting edge Intel Core microarchitecture systems, we also find that our allocation architecture can improve power efficiency by 20% on average.

In this paper we present the beginning of our investigation into exploiting platform heterogeneity and emerging virtualization support to improve the power characteristics of enterprise computing environments. As future work, since modern data centers often have applications such as tiered web services that are network and I/O bound, we will extend our approach to address these types of workloads. Moreover, we also plan to consider how the virtualization abstraction itself can be used to manage dynamic workloads whose behavior or performance demands may vary based upon time or load characteristics. The results presented in this paper support the potential of this area of research for power managing heterogeneous computing systems.

References

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, 2003.

[2] R. Bianchini and R. Rajamony. Power and energy management for server systems. *IEEE Computer*, 37(11), November 2004.

[3] D. Brooks and M. Martonosi. Dynamic thermal management for high-performance microprocessors. In *Proceedings of the 7th International Symposium on High-Performance Computer Architecture (HPCA)*, January 2001.

[4] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle. Managing energy and server resources in hosting centers. In *Proceedings of the 18th Symposium on Operating Systems Principles (SOSP)*, October 2001.

[5] Y. Chou, B. Fahs, and S. Abraham. Microarchitecture optimizations for exploiting memory-level parallelism. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, June 2004.

[6] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, May 2005.

[7] E. N. Elnozahy, M. Kistler, and R. Rajamony. Energy-efficient server clusters. In *Proceedings of the Workshop on Power-Aware Computing Systems*, February 2002.

[8] M. Femal and V. Freeh. Boosting data center performance through non-uniform power allocation. In *Proceedings of the Second International Conference on Autonomic Computing (ICAC)*, 2005.

[9] S. Ghiasi, T. Keller, and F. Rawson. Scheduling for heterogeneous processors in server systems. In *Proceedings of the International Conference on Computing Frontiers*, 2005.

[10] T. Heath, A. P. Centeno, P. George, L. Ramos, Y. Jaluria, and R. Bianchini. Mercury and freon: Temperature emulation and management in server systems. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, October 2006.

[11] T. Heath, B. Diniz, E. V. Carrera, W. Meira Jr., and R. Bianchini. Energy conservation in heterogeneous server clusters. In *Proceedings of the 10th Symposium on Principles and Practice of Parallel Programming (PPoPP)*, 2005.

[12] Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba. Advanced configuration and power interface specification. http://www.acpi.info, September 2004.

[13] C. Isci, G. Contreras, and M. Martonosi. Live, runtime phase monitoring and prediction on real systems with application to dynamic power management. In *Proceedings of the 39th International Symposium on Microarchitecture (MICRO-39)*, December 2006.

[14] R. Kotla, S. Ghiasi, T. Keller, and F. Rawson. Scheduling processor voltage and frequency in server and cluster systems. In *Proceedings of the Workshop on High-Performance, Power-Aware Computing (HP-HPAC)*, 2005.

[15] R. Kumar, D. Tullsen, P. Ranganathan, N. Jouppi, and K. Farkas. Single-isa heterogeneous multi-core architectures for multithreaded workload performance. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, June 2004.

[16] H. Li, C. Cher, T. Vijaykumar, and K. Roy. Vsv: L2-miss-driven variable supply-voltage scaling for low power. In *Proceedings of the IEEE International Symposium on Microarchitecture (MICRO-36)*, 2003.

[17] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making scheduling cool: Temperature-aware workload placement in data centers. In *Proceedings of USENIX '05*, June 2005.

[18] K. Rajamani and C. Lefurgy. On evaluating request-distribution schemes for saving energy in server clusters. In *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, March 2003.

[19] P. Ranganathan, P. Leech, D. Irwin, and J. Chase. Ensemble-level power management for dense blade servers. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, 2006.

[20] W. Zhang. Linux virtual server for scalable network services. *Ottawa Linux Symposium*, 2000.