Exploring Hardware Profile-Guided Green Datacenter Scheduling

Weichao Tang[12], Yu Wang[1], Haopeng Liu[1], Tao Zhang[1], Chao Li[2], and Xiaoyao Liang[1]
[1]*Advanced Computer Architecture Laboratory* and [2]*Sustainable Architectures and Integration Laboratory*
Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
E-mail: {lichao, liang-xy}@cs.sjtu.edu.cn

***Abstract*- Recently, tapping into renewable energy sources has shown great promise in alleviating server energy poverty and reducing IT carbon footprint. Due to the limited, time-varying green power generation, matching server power demand to runtime power budget is often crucial in green datacenters. However, existing studies mainly focus on the temporal variability of the power supply and demand, while largely ignore the spatial variation issue in server hardware. With more complex computing units integrated and the technology scaling, the performance/power variation among nodes and the conservative supply voltage margin of each core can greatly compromise the power matching effectiveness that a green datacenter can achieve. This paper explores green datacenter design that takes into account non-uniform hardware power characteristics. We propose *iScope*, a novel power management framework that can (1) expose architecture variability to the datacenter facility-level scheduler for efficient power matching, and (2) balance the energy usage and lifetime of compute nodes in the highly dynamic green computing environment. Using realistic hardware profiling data and renewable energy data, we show that *iScope* can reduce the energy cost up to 54%, while maintaining fairly balanced processor utilization rate and negligible profiling overhead.**

Keywords- green datacenter; process variation; renewable energy; runtime profiling; power management

I. INTRODUCTION

The exploding cloud service today comes with a price: data centers consume a significant amount of electricity (mainly generated from fossil fuel) and indirectly cause greenhouse gas emissions. To support the ever-increasing server energy needs and minimize negative environmental impact, many recent proposals start to explore the use of renewable energy in datacenters. The continually decreasing prices of green energy have also made such design a proven alternative to existing utility-power-only solutions [1, 2].

In emerging green datacenters, servers demand more judicious power allocation than before. This is because the natural variability of wind/solar power generation, coupled with the fluctuation in computing load, can cause frequent power re-allocation across compute nodes. To maintain a continuous power balance between the supply and demand, green datacenters often suffer recurrent processor per-core power gating [3], load migration between nodes [4], job deferring on server clusters [5, 6], or on/off power state switching on servers [7]. In such a highly dynamic operating environment, even small changes in processor energy consumption or assigned power budget can result in large differences in datacenter efficiency and productivity.

However, existing studies mainly focus on the power variability issue at the datacenter facility level and overlook the efficiency variation in hardware. As we enter the deep silicon regime, process variation (PV) in IC design has become an issue that cannot be ignored, especially in the context of chip-multiprocessors (CMPs) and thousand-node clusters. It is no longer accurate to treat a datacenter as homogeneous system even if all the employed servers are of the same configuration. For example, core to core (C2C) variation has been identified and the maximum difference in core frequencies is estimated to be 20% [8]. CMPs with non-uniform power characteristics can lead to scheduling and energy management problems. In addition, variability in transistor parameters is forcing more conservative design methodologies. To ensure correct operation, high guardbands are added to the supply voltage to account for the worst case. These guardbands, which can be as high as 20% [9], make processors less efficient. Consequently, if the hardware characteristics are ignored during job scheduling, the effectiveness of supply-demand power matching in green datacenters may be severely compromised.

To better utilize the computing resource and renewable energy in green datacenters, we take the first step to investigate a hardware profile-guided power management strategy. Previous works only look at handling the supply-demand power mismatch issue of the entire datacenter system (i.e., macro level) [4-7]. In contrast, we explore the benefits of combining prior art with a "micro level" control that fine-tunes the power allocation across processor cores based on detailed hardware profiling data.

In this paper we propose *iScope*, a novel macro-micro multi-dimensional power management framework for green datacenters. It is designed to automate two key processes:

- *Dynamic hardware scanning.* iScope allows a green datacenter to periodically scan its server nodes and distill crucial processor variability information via software-based functional test. This process has negligible overhead and no additional hardware is required. iScope exposes each processor's process variation and voltage margin characteristics to the datacenter for power optimization purpose.
- *Variation-aware scheduling.* Our framework can smartly allocate power to compute nodes with the awareness of both power budget variability and hardware variation statistics. It enables the datacenter to adjust the number and type of compute nodes to use under different power budget for the sake of lower utility energy consumption, better renewable energy utilization, and balanced processor lifetime.

0190-3918/15 $31.00 © 2015 IEEE
DOI 10.1109/ICPP.2015.10

The rest of this paper is organized as follows. Section 2 introduces the background. Section 3 proposes runtime profiling in green datacenter. Section 4 presents our variation-aware scheduling algorithms. Section 5 describes evaluation methodology followed by Section 6 discussing experiment results. Finally, Section 7 discusses related work and Section 8 concludes this paper.

II. BACKGROUND

A. Power Variation in Green Datacenters

Recent research has demonstrated the benefits of renewable energy powered datacenters. For example, most renewable energy systems (RES) are modular, making it possible to incrementally expend datacenter power capacity with zero carbon emission. The initiation time of RES is often much shorter than that of conventional power plants, which allows for quick deployment of new datacenter facilities. In addition, the price of renewable energy is declining. It has been shown that the price of wind energy could be less than 0.005 USD/kWh in the near future [2]. Given that the global server power demand continues to expand rapidly, tapping into renewable energy can greatly alleviate the escalating energy needs of datacenters. Many companies such as Google, Microsoft, Facebook, and eBay have already started to explore this design option.

One of the biggest challenges in green datacenter design is managing the time-varying gap between renewable power budget and workload power demand. Energy sources like solar and wind can change from full grade to zero within minutes. Heavily relying on the utility power grid and large-scale onsite battery to complement RES has been shown to be inefficient and costly [1, 10]. Therefore, many recent studies have focused on managing renewable power supply variation and datacenter power demand variation at the system level [3-7, 10-13]. However, to our knowledge, none of the prior work ever takes into account detailed processor characteristics in green datacenter design.

B. Efficiency Variation in Processor Nodes

Whereas many recent green datacenter studies have explored the temporal variations in renewable power supply and datacenter demand, they overlook the efficiency variations of the underlying hardware. Imprecise control of the transistor parameters in the IC manufacturing process can lead to process variation (PV) which increases with sub-nanometer technology. PV changes several key transistor parameters including the threshold voltage (V_{th}) and the effective gate length (L_{eff}). These parameters directly affect a transistor's switching speed and leakage power. As a result, the real operational frequency of the processor often scatters around the designed nominal point after chip fabrication. Besides, V_{th} variation causes significant leakage power variations across chips due to their exponential relationship. Intel has reported that PV can cause up to 30% deviation in frequency and up to 20× variation in chip leakage power in high-end processors [14].

Process	6376	6378	6380
Core/Cache (MB)	16/16	16/16	16/16
Nominal Clock (GHz)	2.3	2.4	2.5
Max Clock (GHz)	3.2	3.3	3.4
Price ($)	703	876	1088

Table 1: Three bins of the AMD Opteron 6300 CPU

In addition, C2C variations in multi-/many- core systems arise due to spatially correlated within-die (WID) variation whose chief impact manifests across rather than within cores [15]. This has a profound impact on not only core performance, but also power behaviors. Approximately a 4× variation in leakage power has been shown in a four-core homogenous processor under 65nm [16]. This means that the identically designed processors are no longer the same in terms of efficiency once fabricated. Energy efficiency and its variability have been considered as a major issue on the next 15 years' technology roadmap [17].

Another problem with PV is conservative voltage margin. Traditionally, speed binning process categorizes processors based on their performance, which is helpful to improve chip yield. For example, the AMD Opteron 6300 series processor [18] is a high-end server CPU which has three bins as shown in Table 1. The processors are classified into one of the three bins according to whether it can sustain in a threshold frequency at certain voltage. However, although the processors in the same bins work at the same frequency, the minimum voltage (*Min* V_{dd}) at the nominal clock roughly has 5% variation among cores (detailed in Section 5). This requires the processor to operate with large voltage margins (which is less efficient) to guarantee correctness under worst-case conditions that rarely occur.

To quantify process variation and voltage margin in modern processors, experiments were conducted on four AMD A10-Series A10-5800K quad-core processors [19]. More details about the evaluation are presented in Section 5. The lowest safe voltage at which each core runs reliably in nominal frequency is archived through profiling. The results show that design-identical cores have different *Min* V_{dd}. The *Min* V_{dd} ranges between 1.19V and 1.25V (the nominal voltage is 1.375V). All cores run reliably at voltages that are 9% lower than nominal values. In addition, many features in current CPUs are untapped in certain specific work environment but have a great impact on the supply voltage. Integrated GPU is an example. Our real measurement shows that enabling the integrated GPU can increase the *Min* V_{dd} by 10.3% in an AMD Quad-core processor.

With the rapid growth in the quantity and utilization rate of advanced chips in green datacenters, ignoring PV can cause significant efficiency degradation. The intention of this work is to *provide an initial framework that allows a green datacenter to make informed power management decision based on hardware characteristics.* In Section 3 we first introduce the dynamic hardware scanning strategy of iScope. We then present different variation-aware power allocation algorithms in Section 4.

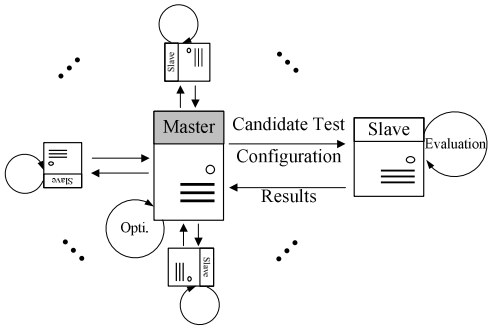


Figure 1: Software-based functional failing test

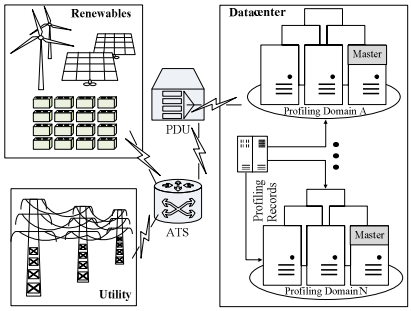


Figure 2: Dynamic hardware scanning architecture

III. DYNAMIC HARDWARE SCANNING

We propose *iScope*, a novel power management framework that can expose the variation characteristics of processors at the system level. *iScope* comprises two key elements: *iScope scanner* and *iScope scheduler*. This section introduces the *iScope scanner* which is a software toolchain that gives today's green datacenter a fairly complete view of the underlying hardware.

A. Software-based Functional Failing Test

We use software-based functional failing test [20] to distill process variation information in green datacenters. It is basically an assembly-language program whose result is functionally incorrect. That is, the misbehavior can be detected by simply checking the result at the end of execution. The required test program can be generated automatically by various algorithms [20, 21]. In contrast to traditional chip functional testing, the software-based functional failing test requires neither an expensive tester, nor any design-for-debug circuit [22]. It exploits the feedback from the examined chips and does not require any information about the underlying microarchitecture.

Another advantage of software-based functional failing test is that it can be easily applied in workplace, as shown in Figure 1. The master computer creates a candidate test and controls the operating configuration (frequency and voltage) of the slave; the slave adjusts to the configuration point and executes test programs. Once the slave executes the program, the most relevant feedback is the configuration threshold of the candidate test. That is, the frequency and voltage when the results cease to be correct. More fine-grained test can be carried out with increased time and energy overheads.

B. Fine-Grained Frequency and Voltage Control

Our framework is based on the frequency and voltage control capability of existing processor microarchitecture. Today, separated clock domain for each core is common for current processors. Normally, one core has a specific PLL to generate customized clock frequency. For example, there is an independent clock generator for each of the eight cores in Xeon EX processor [23]. The same situation can be seen in AMD and IBM serial CPUs. Therefore, it is practical to adjust core frequency in real time.

For the purpose of exploring each core's minimum safe voltage at different frequency bins, voltage regulator (VR) is necessary to adjust processor operating voltage dynamically. There are some studies on off-chip VR, on-chip VR and hybrid VR [24]. Off-chip VR doesn't occupy on-chip power grids and area. It has higher power delivery efficiency, but is not as responsive as on-chip VR (tens of microsecond timescales). In contrast, on-chip VR has much shorter latency (nanosecond timescales) to switch to a new voltage, but it has relatively lower power delivery efficiency and dictates significant amount of chip area. So hybrid scheme is proposed to combine both advantages.

To support per-core voltage control, the processors need power grids and voltage regulators that generate different voltages for each core. Per-core voltage domains in multi-core processors have been suggested [25, 26]. In [25], a cost-effective power delivery technique, on-chip low-dropout (LDO) VRs, is proposed to further decrease the on-chip VR cost and support per-core voltage domains. At the same time, per-core voltage domains can achieve energy savings (>20%) when compared to conventional single power domain. For example, AMD's Griffin processor provides dual-power planes for per-core voltage/frequency control [27]. In the Intel's Itanium II processor, V_{dd} lines are shared by core pairs [28]. It is reasonable to believe that a growing number of the next-generation microprocessors will support per-core voltage and frequency scaling.

C. Runtime Processor Profiling

Our runtime hardware profiling architecture in green datacenters is shown in Figure 2. Idle processors in the datacenter can set to a stable configuration point and act as master/monitor to profile other processors.

We propose an opportunistic hardware runtime profiling. The newly acquired processors are physically installed into the datacenter. Because the processors can operate reliably at the nominal configuration point, it is safe to add these units without affecting the normal operation. Since the server workload changes dynamically, these nodes will be opportunistically profiled especially when the datacenter is idle or under low utilization. In this case, isolating the nodes from normal service will not affect the quality of service (QoS). Once these nodes are separated, a specific

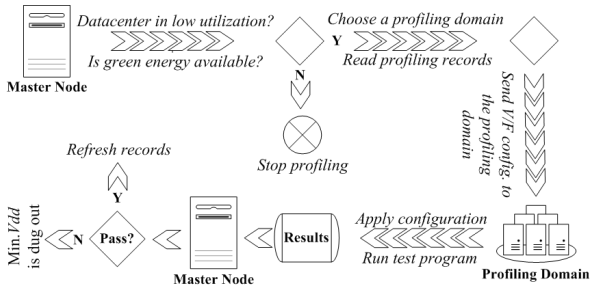


Figure 3: Runtime processor profiling flow

frequency/voltage configuration and a pre-defined software-based functional test or stress test routines are fed into the test processors to figure out the exact chip behavior. Both stress test and software-based functional failing test can check the system stability [20]. The only difference between them is that stress test needs a little more time (10 minutes) than software-based functional failing test (29 seconds). We use stress test to check processor robustness.

In Figure 3 we show the detailed profiling flow of our dynamic profiling. It mainly consists of six stages:

- *When the renewable energy generation is available and datacenter is at low-utilization, executing profiling program on idle processors (master nodes) which work at the stable configuration point. Otherwise, stop profiling.*
- *The profiling program then reads the profiling records and chooses a group of inadequately profiled processors, which constitute a profiling domain. Any given processor only belongs to one specific profiling domain.*
- *Based on the profiling records, the profiling program sends voltage/frequency configuration and stability test program to each core in the profiling domain.*
- *The test processor receives its voltage and frequency configuration and our system adjusts it based on the configuration through hardware drivers. We then execute the stability test program which is stored at a special address in cache.*
- *Once the stability test program finishes, the test processor returns the execution result to the master processor. The result is compared with the pre-defined correct value. Label "pass" or "fail" for each core at corresponding V/F configuration sets based on the comparisons.*
- *Refresh the profiling records. If a "fail" is recorded, lower voltage configurations at the same frequency bins force to "fail". The minimum V_{dd} under this frequency is dug out.*

Fine-grained chip profiling can be achieved using the above control flow. As long as the PLLs and VR provide enough settings, more voltage/frequency configuration point can be tested for each processor. In this case, each core has much more freedom for better energy efficiency.

The scanning data is reported back to the scheduler and stored into its database. This information can be updated at runtime. It allows the scheduler to develop an understanding of the variation map of the underlying processors and various configurations. It is beneficial to expose the chip physical variation to the system software so that the scheduler can optimize the datacenter operations to best compensate for other inherent system variations such as cloud workloads and renewable energy fluctuation.

It is worth pointing out that datacenter can perform on demand profiling. Modern processors integrate many features but not all of them are useful to the cloud operation. For example, if the integrated GPU is not used in the cloud service, the profiling can simply omit checking that part, possibly boosting chip frequency or lowing chip V_{dd}.

In addition, green datacenters should perform the profiling periodically, especially when servers may undergo aggressive and unbalanced power tuning activities (e.g., clock throttling and voltage scaling [3, 12], frequent on/off power cycling [7], etc.). Divergent working conditions and utilization times wear out processors differently, which can redistribute the variations among chips. Periodical profiling is an effective way to timely expose processor variation.

IV. VARIATION-AWARE SCHEDULING

The processor characteristics captured by *iScope scanner* enables a green datacenter to fine-tune its power allocation from a micro perspective. In this section we describe *iScope scheduler* which exploits the hardware profile for improving the power allocation efficiency in green datacenters. The power management problem we consider mainly focus on: when to use and how to use the profiled processors under different renewable energy availability scenarios.

A. Models for Energy and Execution Time

Our scheduler uses build-in power models to estimate processor power during runtime. Our evaluation is relatively conservative since at this stage we mainly focus on the power variation of the processor. If a workload is memory, I/O or network bounded, the energy consumption may outweigh that of a processor. In this case a node-level profiling is necessary if one wants to maximally release the efficiency potential of the datacenter.

In this study we consider both computing power and cooling power. The power consumption P of a CPU can be approximated by the following function [29, 30]:

$$p = \alpha f^3 + \beta , \quad (1)$$

where β stands for static power, α is a CPU-specific constant used for calculating dynamic power, and f is the operating frequency of the processor core.

The energy consumed by the associated cooling system is modeled using a coefficient COP [29, 31], which is the ratio of computing power to cooling power. The study conducted by Greenberg et al. [32] indicated that COP follows normal distribution between [0.6, 3.5]. The total server energy consumption is given by [29]:

$$E_{total} = E_{cooling} + E_{CPU} = (1 + \frac{1}{COP})E_{CPU} \quad (2)$$

The execution time of an application depends primarily on the computational capability of the CPU, which is determined by its frequency. However, the decrease in execution time due to the increase in CPU frequency depends on whether the application is CPU-bound or not. For a completely CPU-bound application, its execution time will be inversely proportional to the change in CPU frequency. The relationship between execution time and CPU frequency is modeled according to [33]:

$$T(f) = T(F_{\max}) \times (\gamma^{CPU} (\frac{f_{\max}}{f} - 1) + 1), \quad (3)$$

where $T(f)$ is the execution time of an application under a specific frequency f and γ is the CPU boundness of the application. In our discrete event-driven simulation, tasks come dynamically with information including requested number of CPUs, CPU boundness, estimated execution time under a specific frequency and the expected deadline for finishing the task. Prior work has shown that such CPU activity traces can provide fairly accurate server-level power prediction [34]. If the scheduler chooses to execute a task with a different frequency than specified, the new execution time can be calculated using the equation above.

B. Scheduling Algorithms

In a datacenter with multi-dimensional variations, a smart scheduler is of paramount importance. Since there is a huge gap in the price between utility power and renewable energy, oftentimes it is optimal to use the renewable energy whenever possible. Furthermore, the scheduler needs to balance the usage of all the processors in the datacenter for an extended life time. Processors exhibit different energy efficiency due to process variation. Efficient processors might be overloaded if not paying attention to their wear out time, causing extra replacement cost.

We have devised five schemes, as listed in Table 2. Each scheme is a combination of a processor profiling strategy and a scheduling algorithm. There are two profiling strategies: *Bin* and *Scan*. The former refers to conventional binning process: processors go through rigorous binning tests in the factory and no additional profiling is performed during operation. *Scan* stands for the case that dynamic profiling is performed in datacenter using our proposed framework. Moreover, we consider three scheduling rules: *Ran*, *Effi* and *Fair*. In *Ran*, workloads are assigned to CPUs randomly. In *Effi*, workloads are always allocated onto a set of available CPUs with the best energy efficiency. In *Fair*, the scheduler attempts to balance the running time of CPUs in an effort to avoid early wear-out, but at the same time tries to improve green energy utilization.

BinRan: In this scheme, processors are categorized into different bins in the factory. The datacenter operates processors strictly according to the manufacturer's binning specifications. No in-cloud profiling is carried out. The scheduler does not consider the variation of different processors. Incoming tasks are randomly assigned to CPUs as long as the processors can meet the deadlines.

Name	Profiling	Scheduling Algorithm
BinRan	No	Random
BinEffi	No	Minimize Energy
ScanRan	Dynamic	Random
ScanEffi	Dynamic	Minimize Energy
ScanFair	Dynamic	Minimize Energy + Balance Utilization

Table 2: Evaluated task scheduling schemes

BinEffi: This scheme also doesn't execute in-cloud profiling process as in *BinRan*. The difference lies in the scheduling policy. This algorithm attempts to minimize the total energy cost with the manufacturer's binning information. For an incoming task, the scheduler always maps it to a set of CPUs with the best power efficiency while respecting the deadline. Different processor bins have different power efficiency. Cloud workloads are highly dynamic with varying requirement for CPUs. Instead of blindly scheduling tasks, this policy takes efficiency variation into account. Nevertheless, since all the processors in the same bin are treated as identical, the scheduler cannot leverage the fine-grained efficiency difference between processors in the same bin, missing the opportunity to improve energy efficiency.

ScanRan: This scheme explores the benefit of in-cloud profiling. Processor performance/power profile is now available for the datacenter. This effectiveness means variations among processors are exposed and label, providing more options for operation. The scheduling policy is still random without considering the efficiency variation. Compared with BinRan, the advantage of this scheme is that two processors in the same bin in BinRan can now be further divided into fine-grained configurations, tracking more closely to each chip's optimal efficiency point.

ScanEffi: This scheme also uses in-cloud profiling to expose processor variation. With the detailed processor profile, the scheduler now can better tailor the overall energy consumption and adapt to the varying workloads and power supply. The scheduler will always schedule the most energy-efficient CPUs for incoming workloads as long as the deadlines can be met. Compared with BinEffi, more flexibility are provided to scheduler and each processor can operate closer to its best efficiency point.

ScanFair (default for iScope)**:** This scheme uses in-cloud profiling. Its scheduling algorithm is designed to address the unbalanced usage of processors in the cloud. Processors wear out much faster with intensive usage. Replenishing early retired CPUs incurs extra charge. To save energy cost, *BinEffi* and *ScanEffi* always pick the most energy efficient processors for task execution, which causes higher burden on those units. This is adverse to the cloud operators because they tend to upgrade the processors to the next generation in a batch instead of replacing individual short-lived processors.

ScanFair is the default configuration in iScope. It seeks a balance between the energy consumption and the processor usage time. In a datacenter powered by both renewable energy and utility power, the scheduler adapts its policy at run time. At the time the renewable energy is low and a large amount of utility power need to supplement, the algorithm always tries to pick the most energy-efficient processors for incoming jobs so as to save the expensive utility power. On the other hand, with abundant renewable energy, the algorithm picks historically least-used CPUs which might be relatively inefficient for job execution to balance the processor usage time. Power consumption is increased in this case but the renewable energy is generally cheaper. However, the efficient CPUs get a chance to take a rest and their life time can be extended. The key concept behind the *ScanFair* scheme is to jointly manage the processor, workload and renewable energy variations for improving green energy utilization while maintaining a balanced processor lifetime.

V. EXPERIMENT METHODOLOGY

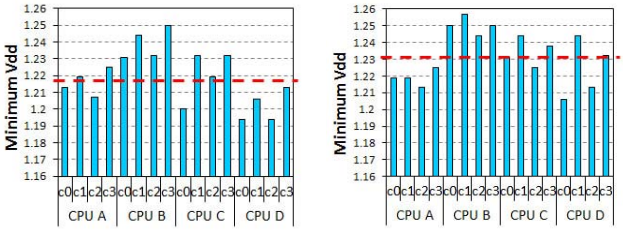
A. Processor Profiling

In this study we use AMD A10-Series A10-5800K processor [19] (each possessing four cores) for profiling performance/power variation. A10-5800K, which nominal frequency is 3.8GHz, has a Radeon HD 7660D integrated GPU. The system is provisioned with 4GB of DDR3 memory and ran the Ubuntu operating system.

We profile four same A10-5800K CPUs (16 cores) with the Mprime stress test program [35], which is a Linux version of prime95. All stress test runs 10 minutes at one voltage and frequency configuration. We label each core "pass" or "fail" based on the stress test result. In our experiment, four A10-5800K processors run at the nominal frequency (3.8GHz). The processor V_{dd} is gradually decreased, while running stress test workload on each core until all cores cannot pass stress test. The lowest safe voltage at which each core runs reliably is recorded.

Firstly, we disable the integrated GPU in the A10-5800K and profile under text-based user graphics. Figure 4(A) shows the *Min* V_{dd} for each core in this case. The *Min* V_{dd} ranges between 1.19V and 1.25V. The average *Min* V_{dd} (red dash line in figure) of 16 cores is 1.219V. This core-level variation outlines the variability in performance/power among servers and the potential to further improve the efficiency by more sophistic scheduling.

Integrated GPU in processors has a significant influence on cores performance/power. We enable the integrated GPU and profile under the Ubuntu graphical user interface (GUI). The integrated GPU frequency is set to 1900MHz. The profiling results are shown in Figure 4(B). When the integrated GPU is enabled, the *Min* V_{dd} ranges between 1.206V and 1.2506V. The average *Min* V_{dd} (red dash line in figure) of 16 cores is 1.232V, 10.25% higher than processors with a disabled integrated GPU.



(A) Integrated GPU disabled (B) Integrated GPU enabled

Figure 4: Real experiment data for Minimum V_{dd} of the AMD four A10-5800K Quad-core processors

B. Model for Processors

The parameters for CPU power in Eq-1 are derived from [30]. We use the analytical model in [36] and assume a Poisson distribution for β with a mean of 65 and a normal distribution for α with a mean of 7.5 and a variation of 0.75. The mean values of α and β refer to [30]. The processor we evaluated can apply 5 dynamic V/F scaling levels with a frequency range from 750 MHz to 2 GHz. All the processors have the same frequency settings but need different voltages and in turn exhibit different power efficiency. For in-factory binning, processors are grouped into 3 bins according to their power efficiency, similar to the AMD Opteron 6300 series. Processors falling into the same bin must apply the same voltage of the worst-case chip in that bin to ensure normal operation. However, with dynamic hardware profiling, every processor can use the optimal voltage according to its own variation.

C. Datacenter Configuration

We model a green datacenter with 4800 CPUs. The cooling coefficient COP in Eq-2 is set to be 2.5, similar to Garg et al.'s approach [29]. We assume the datacenter can operate with both renewable energy and utility power. The datacenter is assumed to be at California where many big internet companies are located, with the utility power price of 0.13 USD/kWh.

We use wind as the renewable energy supply because it is cheap and widely used in large scale facilities [2]. The wind power traces come from the Wind Integration Datasets [37] of the National Renewable Energy Laboratory. The datasets are sampled every 10 minutes from commercially prevalent wind turbines. In the original trace, the available power is much more than what we need for 4800 CPUs, thus we simply scale down to 3.5% of the original level.

Our experiments try to maximally utilize the renewable energy. If the renewable power is not enough to run all the required processors at full speed, DVFS is applied to reduce the frequency and power demand. We stop lowering the frequency when some tasks are facing violation of their deadlines. If the renewable power is still not enough at that time, we will supplement utility power for QoS considerations. Utility power costs more than renewable energy and in this case, higher electricity bill is expected.

D. Workload Configuration

We use the LLNL Thunder workload traces from the well-established Parallel Workload Archive (PWA) [38]. PWA is a collection of traces from real clusters or production systems. The evaluated trace contains logs from a large Linux cluster (with 4096 processors in total) installed at the Lawrence Livermore National Laboratory. Parameters for workload include submit time, requested number of CPUs, runtime, CPU time, etc.

We adopt the same idea as in [29] to assign deadlines for each task into two urgency classes: *High Urgency* (HU) and *Low Urgency* (LU). HU workloads have more critical deadline requirement thus should be treated in higher priority. HU follows a normal distribution with a mean of 4× the nominal execution time and a variance of 2. LU follows a normal distribution with a mean of 12 and a variance of 2. We also adjust the arrival rate of tasks to simulate the overall loading of the datacenter. For example, an arrival rate of 5X indicates the adjusted task submit time is 20% of the origin setting. This means that new tasks will come more frequently.

VI. EVALUATION

In this section we evaluate the impact *iScope* on green datacenters. We present a series of case studies and analyze the proposed schemes under different operating scenarios.

A. Utility-Power-Only Design

We first evaluate conventional datacenters powered by utility grid only, as shown in Figure 5(A). By comparing the utility energy consumption under different percentage of HU workloads, we can conclude the following results. **(1)** Hardware efficiency can have a significant impact on datacenter energy consumption. In both plots, *Effi* schemes are always better than Ran schemes since *Effi* chooses the most power efficient processors for execution whenever possible while *Ran* simply picks random processors from the resource pool. **(2)** Variation-aware power management provides further energy savings. In Figure 5(A), *Scan* schemes outperform *Bin* schemes by roughly 10% since our dynamic fine-grained profiling enables individual processor to operate at the optimal power efficiency level.

Figure 5(B) compares the utility energy consumption of the five schemes under different job arrival rates. *Ran* schemes consume relatively stable energy with rising job arrival rate because our experiment assumes adequate processors for the incoming jobs. As long as the total number of jobs does not change, varying the job arrival rate will not affect the energy consumption for a random scheme. However, the energy consumption goes up for *Effi* schemes with higher job arrival rate. This is because a growing number of energy-inefficient CPUs have to be chosen if more jobs are coming in a short time, reducing the optimization effectiveness of the *Effi* schemes. We have the similar observation in Figure 5(A), where *Effi* schemes consume relatively higher energy with more HU jobs.

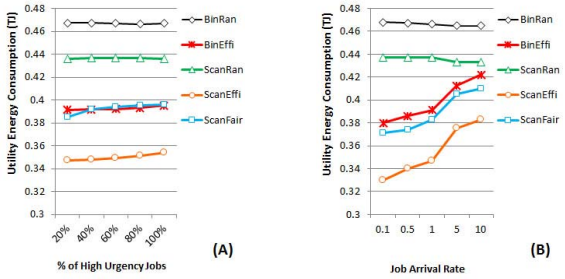


Figure 5: The utility energy consumption vs. % of HU and job arrival rate for the five scheduling schemes

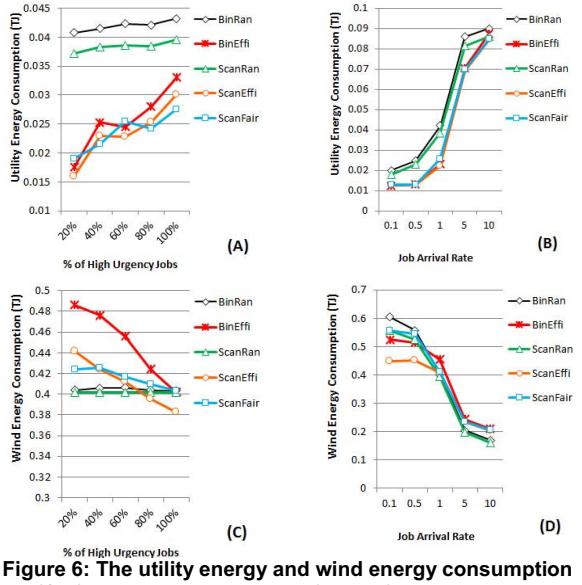


Figure 6: The utility energy and wind energy consumption vs. % of HU and job arrival rate for the five schemes

B. Utility Power and Wind Energy

Different from previous experiment, we integrated wind energy to the datacenter in this case study. Utility energy serves as a supplement when wind energy is inadequate.

Figure 6(A) and Figure 6(C) compares the utility and wind energy consumption under different percentage of HU jobs. With more HU jobs, we observe that *Effi* schemes tend to use less wind energy but more utility energy. *Effi* schemes always try to assign workloads onto energy-efficient processors. Tasks can be queued up at the energy-efficient processors as long as the deadlines are not violated. Therefore some efficient processors can have a long queue filled with tasks waiting for execution. However, with the increasing percentage of HU jobs having shorter deadlines, *Effi* schemes have to compromise and assign workloads onto more processors including some inefficient ones. As a consequence, workloads are executed with higher parallelism and the total execution time is reduced. Wind energy is reduced with less execution time. In the low percentage HU case, jobs can be gradually executed on the efficient processors fully leveraging the wind power while making the minimum usage of the utility power. With

higher HU, jobs need to be finished much sooner and the utility power budget has to be raised to power up more processors. However for the *Ran* schemes, workloads are randomly assigned without "queueing" phenomenon so there is no obvious energy difference with higher HU rate.

Figure 6(B) and Figure 6(D) present the results for utility and wind energy consumption with different job arrival rate. All the schemes tend to use less wind energy and more utility energy for higher job arrival rate. We use a constant number of jobs in the experiments. A higher job arrival rate leads to a larger number of jobs running simultaneously and consequently a shorter completion time of all workloads. As discussed above, more parallel jobs require more processors being powered up and in turn, consume larger amount of utility energy. Wind energy consumption is reduced because of the shorter task completion time. For most cases *Effi* outperforms *Ran* in terms of utility energy consumption, demonstrating the benefit of efficiency-aware scheduling. However, with a very high job arrival rate, there is no obvious difference between these schemes. This is because the large number of parallel jobs running simultaneously in the datacenter will inevitably require the powering up of energy-inefficient processors, reducing the advantage of *Effi* schemes.

C. *Power Trace and Energy Cost*

Figure 7 demonstrates the real time power trace for *ScanRan*, *ScanEffi* and *ScanFair*. The figures are generated by sampling through the working process every 350 seconds. *ScanRan* assigns tasks randomly. It works relatively well when wind energy is sufficient but loses efficiency when wind energy is low. As shown in Figure 7(A), *ScanRan* consumes more utility power compared to other schemes when the wind power fades away. High utility power generates excessive electricity cost. *ScanEffi*, on the other hand, always makes use of the most efficient processors so that its power consumption is minimized (especially when wind power is low). It saves considerable amount of utility energy than the *ScanRan* scheme. However, *ScanEffi* does not make full use of the wind energy when it is sufficient as can be seen in Figure 7(B). Its trace cannot fit into wind power at high levels. *ScanFair* tries to maximally utilize the wind energy when wind power is high by using historically least-used CPUs which are more likely to be inefficient. *ScanFair* saves utility energy when wind power is low by using efficient processors. In Figure 7(C), *ScanFair* can keep up its pace with the change of wind power by smartly switching between efficient and inefficient CPUs.

In this study, iScope helps a green datacenter improve utility power efficiency and renewable power utilization, thereby cutting energy cost. Figure 8 compares the energy cost with respect to different schemes. The price of utility energy is 0.13 USD/kWh [29] and wind energy costs 0.05 USD/kWh [39]. In the case of no wind energy, the energy costs of *BinEffi*, *ScanEffi* and *ScanFair* are less than *BinRan* and *ScanRan*. This shows the importance of variation-aware

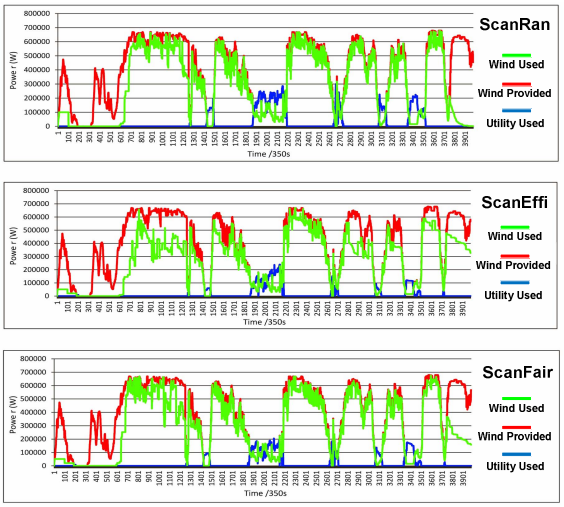


Figure 7: The power trace of three *Scan* schemes

scheduling. *ScanEffi* reduces the cost by 9% over *BinEffi*, proving the effectiveness of the in-cloud profiling. Overall, *ScanFair* achieves 54% energy cost savings over *BinRan*. *ScanEffi* incurs the lowest cost among all the schemes due to its high green energy utilization. If the cost of wind energy continues to decline (e.g., 0.005USD/kWh [2]), green datacenters can further reduce their power cost. In general, *ScanFair* could achieve 30.7% savings on energy (wind & utility) cost over *BinRan*.

D. *Balancing Processor Life Time*

Another key benefit of iScope is that it allows green datacenters to balance processor aging based on the hardware profiling information. Figure 9 shows the variance of processor utilization time in the datacenter. In the figure, we vary the strength of wind speed. SWP stands for standard wind power generation, which is the baseline volume of the wind power. 1.2*SWP means the wind power is amplified by 1.2, providing more renewable energy into the datacenter. We sweep the factor from 1 to 1.8.

We find that the variances of processors utilization time of *Effi* schemes are much higher than others. This is because *Effi* always prioritizes the execution on most energy-efficient processors, resulting in significant imbalance in processor runtime within a datacenter. Such a large variance poses huge burden to some processors and their life time can be greatly reduced. In contrast, *Ran* schemes have the lowest variances due to its random nature, meaning that all available chips get the same chance of being scheduled.

For the design of *ScanFair* (the default configuration for iScope), we aim at balancing the usage of each processor as well as improving renewable energy utilization and saving utility energy consumption. As a result, we observe a relatively lower variance for this scheme. Interestingly, the variance decreases when wind power increases. The reason is that large renewable energy alleviates the constraints for

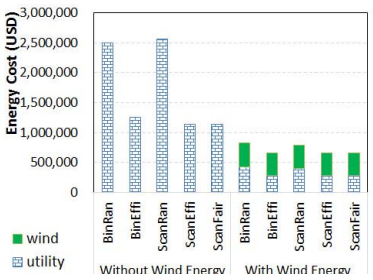


Figure 8: The total energy cost under different task scheduling schemes

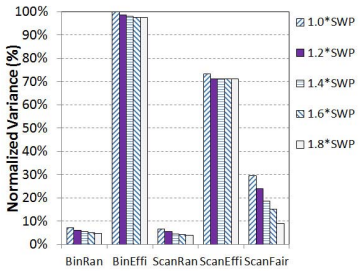


Figure 9: The variance of processor utilization time for five schemes

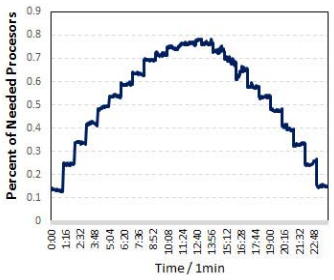


Figure 10: The required number of processors each minute

power consumption in the *ScanFair* algorithm, biasing it to the fairness consideration of the processor usage time. Figure 9 demonstrates the effectiveness of *ScanFair* in balancing the processor life time.

E. Profiling Overhead

There are two kinds overhead for in-cloud profiling in our system: 1) some nodes cannot provide service when profiling, and 2) the energy cost associated with them.

To measure the impact of profiling overhead on datacenter service, we monitor the required number of nodes every minute. In Figure 10, the Y-axis is the percent of required processor number (total available processor is 1024). As we can see, datacenter service request has strong time-related characteristic. The time that required processor less than 30% accounts for 27.2% time in one day, which is enough for 10 minutes stress test program not to say the 29 second software-based functional failing test program [20]. It is also important to mention that the free time is successive not discrete. Moreover, processors can be profiled dispersedly to minimize the influence on quality of service. Therefore, it is practical for datacenter to implement profiling without affecting normal service.

To understand the energy overhead, all processors are set to 115W (the maximum TDP of the AMD Opteron 6300 series) at different voltage and frequency configurations. We profile the processor stability with the stress test running 10 minutes at five frequency bins and ten voltage values. The overall profiling energy cost for 4800 processors in all configuration points is 230 USD using renewable energy. Even using utility power, the overall profiling energy cost is 598 USD. If using software-based functional failing test in [20], only 29 seconds is needed to run the test program. In this case, the profiling cost is 11.2 USD using renewable energy or 28.9 USD using utility power. This is negligible for datacenters.

VII. RELATED WORK

Prior studies either focus on microarchitecture-level designs to reduce processor energy, or system-level power management to better utilize green energy in datacenters.

Process variation affects the frequency and power of fabricated chips [40-42]. In [40], 430 processors in 65nm technology exhibit 25% difference in maximum frequency, and 3X difference in ring oscillator (RO) leakage. Miller et al. [43] proposed a framework for dynamically re-balancing performance heterogeneity caused by process variation and application imbalance in low-voltage chips. Liang et al. [44, 45] presented variation-tolerant circuits and post-silicon tuning techniques for both logic and memory. In contrast to these work, chip characteristics under process variation are identified through dynamic profiling in the datacenter.

Variation-aware job scheduling algorithms have been discussed in [15, 46, 47]. Ndai et al. [15] devised a low-overhead design technique that sets the operating frequency based on the faster units and allows more cycles for the slower units. Teodorescu et al. [46] proposed variation-aware scheduling algorithms to save power and improve throughput. Raghunathan et al. [48] proposed a framework which is able to optimally mapping threads to a subset of cores with different operating frequencies. In this work we consider the using of the system software to exploit physical variations in the green datacenter environment, while the focuses of prior studies are inside the microprocessor.

Many previous studies consider the performance and power consumption in datacenters [49-53]. Soundararajan et al. [50] presented that the management workload from heavy network and disk I/O workflows must be factored into the design of the virtualized datacenter. Reddi et al. [51] concluded that reducing platform power associated with the peripheral components is essential. Kontorinis et al. [52] presented distributed per-server UPSs that offers energy during power spikes. Abts et al. [53] proposed several ways to design a datacenter network whose power consumption is more proportional to the amount of traffic it is moving. Pahlavan et al. [47] discussed server placement and task assignment to minimize datacenter energy consumption with leakage variation. Differently, we are considering the joint effect among chip variation, workloads dynamics and the renewable energy fluctuation in a datacenter.

To cap carbon footprint and reduce energy cost, several recent studies have explored renewable energy powered datacenters. For example, Li et al. [4] coordinated the use of renewable energy and conventional energy in datacenters to reduce the energy cost. Goiri et al. [5, 6] propose to defer data-processing task based on the availability of green energy. Efforts have been made to build green datacenter prototype in [1, 11]. However, to the best of our knowledge, none of the prior green datacenter designs ever consider the detailed characteristics of processors [1, 3-7, 10-13, 54, 55]. By taking a macro-micro multi-dimensional approach, we

can further reduce a green datacenter's dependence on utility energy, increase green energy utilization, while at the same time maintaining a balanced usage of processors.

VIII. CONCLUSION

There is a growing trend towards designing energy-efficient green datacenters. Prior works in this context largely ignore the physical variations of hardware, and therefore miss the opportunity to further improve efficiency. Worse, with the rapid growth in the quantity and utilization rate of compute nodes in datacenters, hardware variation can become a hidden issue that decreases the cost-effectiveness. Through a hardware profile-guided scheduling, we show that existing green datacenters have the potential to further reduce up to 54% energy cost with little overhead while still maintaining a balanced processor lifetime.

IX. ACKNOWLEDGEMENT

This work is partly supported by the National Natural Science Foundation of China (No. 61202026 and No. 61332001) and Program of China National 1000 Young Talent Plan. Chao Li is also supported in part by Shanghai Jiao Tong University New Faculty Start-up Funding and SJTU-MSRA Faculty Award.

REFERENCES

[1] C. Li et al., "Enabling datacenter servers to scale out economically and sustainably. *Int. Symp. on Microarchitecture*, 2013
[2] L.A. Bird et al., "Renewable energy price-stability benefits in utility green power programs", *Technical Report, NREL/TP-670-43532*, 2008
[3] C. Li et al., "SolarCore: Solar energy driven multi-core architecture power management", *Int. Symp. on High-Performance Computer Architecture*, 2011
[4] C. Li et al., "iSwitch: Coordinating and optimizing renewable energy powered server clusters", *Int. Symp. on Computer Architecture*, 2012
[5] I. Goiri et al., "GreenSlot: scheduling energy consumption in green datacenters", *Supercomputing Conference*, 2011
[6] I. Goiri et al., "GreenHadoop: Leveraging green energy in data-processing frameworks", *ACM EuroSys*, 2012
[7] N. Sharma et al., "Blink: Managing server clusters on intermittent power", *ASPLOS*, 2011
[8] E. Humenay et al., "Impact of process variations on multicore performance symmetry", *DATE Conf.*, 2007
[9] V.J. Reddi et al., "Voltage smoothing: Characterizing and mitigating voltage noise in production processors via software-guided thread scheduling", *Int. Symp. on Microarchitecture*, 2010
[10] C. Li et al., "Managing green datacenters powered by hybrid renewable energy systems", *Int. Conf. on Autonomic Computing*, 2014
[11] I. Goiri et al., "Parasol and GreenSwitch: Managing datacenters powered by renewable energy", *ASPLOS*, 2013
[12] C. Li et al., "Enabling distributed generation powered sustainable high-performance data center", *Int. Symp. on High-Performance Computer Architecture*, 2013
[13] R. Singh et al., "Yank: Enabling green data centers to pull the plug", *USENIX Symposium on Networked System Design and Implementation*, 2013
[14] S. Borkar et al., "Parameter variations and impact on circuits and microarchitectures", *Design Automation Conf.*, 2003
[15] P. Ndai et al., "Within-die variation-aware scheduling in superscalar processors for improved throughput", *IEEE Transactions on Computers, vol. 57, no. 7, pp. 940–951*, 2008.
[16] K. Meng et al., "Modeling and characterizing power variability in multicore architectures", *Int. Symp. on Perf. Analysis of System and Software*, 2007
[17] ITRS, "International technology roadmap for semiconductors overview", in 2011 Update, URL: http://www.itrs.net/ reports.html. ITRS, 2011.
[18] "AMD Opteron 6300 series processor". URL: http://www.amd.com/en-us/products/server/opteron/6000/6300
[19] AMD A10-series processor family". http://www.amd.com/us/products/ desktop/pages/consumer-desktops.aspx, 2012.
[20] E. Sanchez et al., "Automatic generation of software-based functional failing test for speed debug and on-silicon timing verification", *Int. Workshop on Microprocessor Test and Verification*, 2011
[21] S. Natarajan et al., "Path coverage based functional test generation for processor marginality validation," *Int. Test Conference*, 2010
[22] J. Zeng et al., "On correlating structural tests with functional tests for speed binning of high performance design," *Int. Test Conference*, 2004
[23] S. Rusu et al., "A 45 nm 8-core Enterprise Xeon Processor," *Int. Solid-State Circuits Conference*, 2009
[24] G. Yan, et al., "Agileregulator: A hybrid voltage regulator scheme redeeming dark silicon for power efficiency in a multicore architecture", *Int. Symp. on High Perf. Computer Architecture*, 2012
[25] H. Ghasemi et al., "Cost-effective power delivery to support per-core voltage domains for power-constrained processors", *Design Automation Conf.*. 2012
[26] W. Kim et al., "System level analysis of fast, per-core dvfs using on-chip switching regulators", *Int. Symp. on High Perf. Computer Architecture*, 2008
[27] "AMD turion x2 ultra dual-core processor". http://www.amd.com/us/ infrastructure/processors/ turion-x2/Pages/turion-x2-mobile-features.aspx, 2012
[28] R. Riedlinger et al., "A 32nm 3.1 billion transistor 12-wide-issue itanium processor for mission-critical servers", *Int. Solid-State Circuits Conf.*, 2011
[29] S.K. Garg et al., "Environment-conscious scheduling of hpc applications on distributed cloud-oriented data centers", *J. of Parallel and Distributed Computing, vol. 71, no. 6, pp. 732–749*, 2011.
[30] L. Wang et al., "Efficient power management of heterogeneous soft real-time clusters", *Real-Time Systems Symposium*, 2008
[31] Q. Tang et al., "Thermal-aware task scheduling to minimize energy usage of blade server based datacenters", *IEEE Int. Symp. on Dependable, Autonomic and Secure Computing*, 2006
[32] S. Greenberg et al., "Best practices for data centers: Lessons learned from benchmarking 22 data centers," *ACEEE Summer Study on Energy Efficiency in Buildings*, 2006
[33] C.H. Hsu et al., "The design, implementation, and evaluation of a compiler algorithm for CPU energy reduction", *ACM SIGPLAN Notices, vol. 38, no. 5, pp. 38–48*, 2003.
[34] P. Ranganathan et al., "Ensemble-level power management for dense blade servers", *Int. Symp. on Computer Architecture*, 2006
[35] "Mprime stress test," http://www.mersenne.org/freesoft, 2013.11
[36] R. Teodorescu et al., "Varius: A model of parameter variation and resulting timing errors for microarchitects", *in Workshop on Architectural Support for Gigascale Integration, in conjunction with ISCA*, 2007
[37] "Western wind and solar integration study". http://www.nrel.gov/electricity/transmission/western_wind.html, 2009.
[38] D. Feitelson, "Parallel workloads archive," http://www.cs.huji.ac.il/labs/parallel/workload/, 2009.
[39] http://www.awea.org/Resources/Content.aspx?ItemNumber=5547
[40] W. Wang et al., "Statistical prediction of circuit aging under process variations", *IEEE Custom Integrated Circuits Conference*, 2008
[41] J. Tschanz et al., "Variation-tolerant circuits: Circuit solutions and techniques", *Design Automation Conference*, 2005
[42] L. Zhang et al., "Process variation characterization of chip-level multiprocessors", *Design Automation Conference*, 2009
[43] T.N. Miller et al., "Booster: Reactive core acceleration for mitigating the effects of process variation and application imbalance in low-voltage chips", *Int. Symp. on High Performance Computer Architecture*, 2012
[44] X. Liang et al., "Process variation tolerant 3T1D-based cache architectures", *Int. Symp. on Microarchitecture*, 2007
[45] X. Liang et al., "Revival: A variation tolerant architecture using voltage interpolation and variable latency", *Int. Symp. on Computer Architecture*, 2008
[46] R. Teodorescu, et al., "Variation-aware application scheduling and power management for chip multiprocessors", *ACM SIGARCH Computer Architecture News, vol. 36, no. 3, pp. 363–374*, 2008.
[47] A. Pahlavan et al., "Variation-aware server placement and task assignment for data center power minimization", *IEEE Int. Symp. on Parallel and Distributed Processing with Applications*, 2012
[48] B. Raghunathan et al., "Cherry-picking: exploiting process variations in dark-silicon homogeneous chip multi-processors", *DATE Conf.*, 2013
[49] S. Rivoire et al., "Joulesort: a balanced energy-efficiency benchmark", *ACM SIGMOD Int. Conf. on Management of Data*, 2007
[50] V. Soundararajan et al., "The impact of management operations on the virtualized datacenter", *Int. Symp. on Computer Architecture*, 2010
[51] V.J. Reddi et al., "Web search using mobile cores: quantifying and mitigating the price of efficiency", *Int. Symp. on Computer architecture*, 2010.
[52] V. Kontorinis et al., "Managing distributed UPS energy for effective power capping in data centers", *Int. Symp. on Computer Architecture*, 2012
[53] D. Abts et al.,"Energy proportional datacenter networks", *Int. Symp. on Computer Architecture*, 2010
[54] C. Li et al., "Characterizing and analyzing renewable energy driven data centers", *SIGMETRICS*, 2011
[55] C. Li et al., "Towards sustainable in-situ server systems in the big data era", *Int. Symp. on Computer Architecture*, 2015