iSwitch: Coordinating and Optimizing Renewable Energy Powered Server Clusters

Chao Li, Amer Qouneh, and Tao Li

Intelligent Design of Efficient Architectures Laboratory (IDEAL)
Department of Electrical and Computer Engineering, University of Florida
{chaol, aqouneh}@ufl.edu, taoli@ece.ufl.edu

Abstract — *Large-scale computing systems such as data centers are facing increasing pressure to cap their carbon footprint. Integrating emerging clean energy solutions into computer system design therefore gains great significance in the green computing era. While some pioneering work on tracking variable power budget show promising energy efficiency, they are not suitable for data centers due to lack of performance guarantee when renewable generation is low and fluctuant. In addition, our characterization of wind power behavior reveals that data centers designed to track the intermittent renewable power incur up to 4X performance loss due to inefficient and redundant load matching activities. As a result, mitigating operational overhead while still maintaining desired energy utilization becomes the most significant challenge in managing server clusters on intermittent renewable energy generation. In this paper we take a first step in digging into the operational overhead of renewable energy powered data center. We propose iSwitch, a lightweight server power management that follows renewable power variation characteristics, leverages existing system infrastructures, and applies supply/load cooperative scheme to mitigate the performance overhead. Comparing with state-of-the-art renewable energy driven system design, iSwitch could mitigate average network traffic by 75%, peak network traffic by 95%, and reduce 80% job waiting time while still maintaining 96% renewable energy utilization. We expect that our work can help computer architects make informed decisions on sustainable and high-performance system design.*

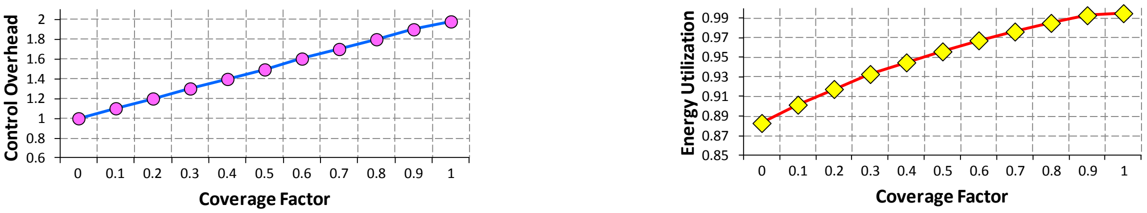
1. Introduction

Today, cloud computing is redefining the IT infrastructure. Data centers have become essential to the operation of businesses, academic, and governmental institutions. Nevertheless, the power-provisioning problem is challenging data center designers, as the environmental impact of IT becomes a growing concern worldwide. It has been shown that worldwide data centers run the risk of doubling their energy consumption every 5 years [1]. In a recent report on the carbon impact of cloud computing, the environmental activist group Greenpeace called on data center operators to make renewable energy a priority as more data centers are being built to meet the cloud needs [2]. In addition, the government also imposes a “carbon tax” on energy-hungry IT companies while giving federal tax credit (e.g., 30% to the total cost) for using renewable energy. Consequently, there has been an increasing push towards the vision of renewable energy powered sustainable computer system design.

In this study, we investigate an emerging data center design scheme that integrates on-site renewable energy sources into the data center infrastructure. Such a design scheme has recently drawn considerable attention as the IT industry is starting to assume responsibility for supporting long-term computing system sustainability. Internet giants, such as Google, Microsoft and Yahoo! all power part of their data centers using renewable energy resources. Using grid utilities as backup, many Web Hosting service providers power their data centers with on-site renewable energy sources as well [3, 4, 5, 6]. With those green computing initiatives, each data center is able to eliminate nearly 20,000 lbs or more carbon dioxide emission per year [3, 4].

While many research efforts are focused on reducing idle server power [7], lowering provisioned power capacity [8], and optimizing power allocation [9, 10, 11], designing renewable energy powered data centers is still challenging and requires careful exploration. Due to the intermittent nature of renewable energy, existing designs typically use on-site renewable generation to compensate part of their total data center power requirement. When renewable energy contributes a large portion of the load power demand (e.g., > 15%), variations in renewable power supply have a significant impact on load operation [12]. Existing power management schemes miss the opportunity to harvest renewable generation since they typically assume a fixed and predictable power supply and cannot handle the power variation gracefully. Recent proposals leverage load tuning mechanisms (e.g. DVFS, CPU power states) to track the renewable power variation [13, 14] but incur unnecessary load tuning activities. Over-tuning of the load power degrades system response time and provides very limited efficiency return.

Figure 1 shows the tradeoff we found between power tuning overhead and energy utilization. The evaluated system tracks the renewable power budget whenever renewable generation decreases (to avoid brownout); it tracks the power supply surge with a pre-defined coverage factor (CF). As shown in Figure 1-a, compared to no-tracking (i.e., CF=0), always tracking the power variation (i.e., CF = 1) increases the load tuning overhead by 2X. Nevertheless, the range of energy utilization return is less than 15%, as shown in Figure 1-b.



(a) Normalized control overhead

(b) Average energy utilization

Figure 1: Power management overhead vs. renewable energy utilization. Always tracking the power surge shows less than 15% energy utilization improvement but experiences 2X control overhead, which affects execution latency, downtime, and efficiency. Both figures show average value across four different renewable energy sites which are detailed in Section 6

In this paper, we explore the design tradeoffs between energy utilization and load tuning overhead in renewable energy driven computing systems. We propose *iSwitch*, a power management scheme that maintains a desirable balance between renewable energy utilization and data center performance. The novelty of our design is two-fold. First, *iSwitch* is a supply-aware power management scheme. It applies the appropriate power management strategy for wind variation scenarios to achieve the best design tradeoff. Second, *iSwitch* has a built-in supply/load cooperative optimization mechanism that is able to minimize the performance degradation due to load power tuning overhead while still maintaining high renewable energy utilization and low cost. Although we describe our design in the context of wind power supply, it applies to a variety of intermittent renewable energy sources.

This paper makes the following contributions:

- **Design:** We propose *iSwitch* control architecture, an application-independent hierarchical load tuning scheme that leverages load migration to best utilize the renewable energy generation.
- **Characterization:** Our characterization of renewable power variability and data center load fluctuation reveals that power tracking can be done in a less frequent, light-weight manner. In this case, we can significantly reduce the load tuning overhead with negligible efficiency degradation.
- **Optimization:** We propose supply/load cooperative optimization that not only avoids redundant load tuning activities invoked by severe renewable power supply variation, but also minimizes unnecessary power control activities invoked by stochastic data center load fluctuation. Compared to the state-of-the-art renewable energy-driven designs, *iSwitch* could reduce job waiting time by 80%, mitigate average network traffic by 75%, and rush hour traffic by 95%. Moreover, *iSwitch* still maintains 96% of the energy efficiency.

The rest of this paper is organized as follows: Section 2 provides background and motivation. Section 3 characterizes wind power variation. Section 4 presents an overview of *iSwitch* and its control architecture. Section 5 introduces the two key constituents of *iSwitch* optimization. Section 6 describes our evaluation framework. Section 7 presents experimental results. Section 8 discusses related work and Section 9 concludes the paper.

2. Background and Motivation

Renewable energy supply (RES) is drawing growing attention in today's IT industry. In this section, we describe the motivations of renewable energy driven design and discuss why conventional solutions limit the development of sustainable IT.

2.1. Renewable Energy Driven Design

In spite of the intermittent nature of renewable energy sources, designing renewable energy powered data centers has many benefits beyond low carbon footprint. For instance, renewable power supplies are highly modular in that their capacity can be increased incrementally to match the gradual load growth [15, 16]. This greatly reduces the over-provisioning loss of a data center since it takes a long time for the server load to catch up with the upgraded provisioning capacity. In addition, the latency between initiation and construction (a.k.a. construction lead-time) of renewable generation is significantly shorter than that of conventional power plants, reducing the financial and regulatory risks [15]. Moreover, the price and availability of renewable resources remain stable, simplifying long-term planning for IT companies [15, 17].

2.2. Problem with Conventional Solutions

Conventionally, extra-scale battery farms can be used to regulate renewable power supply fluctuation. Such approach requires additional large capital investment and is not energy-efficient: the round-trip energy loss of batteries ranges between 5%~25% [18]. Furthermore, frequent charging and discharging the battery accelerates its aging and quickly wears it out [19]. This will further increase the environmental burden (i.e. recycling problem) and the downtime for maintenance.

Alternatively, feedback mechanisms such as net-metering [20] directly connects on-site renewable energy source to the local utility grid to gain high power provisioning availability. Nevertheless, net-metering is still in its infancy stage and aggressively relying on it can be hazardous to the utility operation. This is because grid operators are forced to switch their power stations frequently between operation and spinning standby modes to meet the unexpected feedback power surge. In addition, to ensure stability, the maximum renewable power penetration of the transmission line also has a limit [15]. In the foreseeable future, these problems can be a major hurdle for going green. Waiting for the grid to be "smart" will unnecessarily delay the long-term goal of sustainable IT.

The motivation of this work is that the IT facility itself can be an enabler of sustainability and efficiency. Instead of adding additional power provisioning units and server clusters, our technique leverages existing data center infrastructure and load migration mechanisms to manage the time-varying renewable power. As an alternative to entirely relying on power grid or energy storage, we exploit intelligent data center self-tuning to manage the IT power provisioning. By reducing the dependence on large-scale batteries and utility power grids, our design improves data center sustainability with low cost while providing the operators with more active control of their server clusters.

3. Implications of Wind Power Variability and Intermittency

While the importance of managing renewable energy intermittency is recognized by recent studies [13, 14], characterizing power variation behavior is still an unexplored area in the context of data center design. In this study we primarily focus on wind power variation since wind energy is cheaper and is also the most widely used renewable power technique for large scale facilities [15]. In this section, we first discuss wind power characteristics. We then demonstrate three critical power management problems in wind energy driven data centers and the associated optimization opportunities.

3.1. Wind Power Characteristics

A wind turbine generates electrical power by extracting kinetic energy from the air flow. While operating, the turbine converts wind energy to mechanical power through a rotor with multiple blades. Figure 2 shows the output characteristics of a GE wind turbine, whose power curve is divided into three regions by the designated operating wind speeds. The cut-in speed is the minimum speed at which the rotor and blade starts to rotate. The cut-off speed is the wind speed at which the turbine ceases its generation and shuts down for protecting the blade assembly.

In Figure 2, we refer to the three regions as intermittent power outage period (Region-I), variable power generation period (Region-II) and stable power generation period (Region-III), respectively. In Region-I, wind power is intermittently unavailable because the wind speed is either too low or too high. In Region-II, the mechanical power delivered to the turbine generator is given by $p = 0.5\rho A v^3 C$ [15], where ρ is the air density, A is the swept area of the blades, v is the wind speed and C is called power coefficient factor. In Region-III, the wind turbine operates at its designated rated power.

Wind power has the highest variability in Region-II. To understand this, we show the probability distribution of wind speed in Figure 3. The variations in wind speed are typically described by the Weibull distribution [15]:

$$f(v) = \left(\frac{k}{c}\right)\left(\frac{v}{c}\right)^{k-1} e^{-\left(v/c\right)^k}, v \in [0, \infty) \tag{1}$$

In Equation-1, k is the shape parameter, c is the scale parameter and v is the wind speed. At most wind farm sites, the wind speed has the Weibull distribution with k = 2, which is specifically known as the Rayleigh distribution [15]. As shown in Figure 3, the Rayleigh distribution function in Region-II is not monotonic. In this region, wind speed has equally high possibilities at a wide range of values. As a result, the wind turbine is more likely to incur time-varying wind speed in Region-II. In addition, the wind turbine output is a steep curve due to the cubic relation between wind power and wind speed. In this case, a small change of the wind speed can lead to large wind generation fluctuation. Therefore, the renewable power variability is typically significant in Region-II.

3.2. Power Management Regions

Figure 4 shows real traces of wind power supply and data center power demand. It illustrates the aforementioned three power generation scenarios, namely, intermittent wind power outage period (Region-I), low renewable generation with frequent fluctuation (Region-II), and full renewable generation with relatively stable output (Region-III). Similar region partition method is also applicable for intermittent renewable energy sources such as solar power and tidal power. In the following paragraphs we will discuss data center design considerations for each region.

Region-I: tune wisely. During the low generation period (i.e. Region-I), it is wise to shift data center load from renewable energy supply side to utility power. To tune the load power footprint, existing practices either put servers into low power states or apply power cycling techniques [14] on the hardware. Although these approaches show impressive power control capability, they sacrifice the computing throughput. In addition, it typically takes a long time (about tens of minutes as we observed) for the renewable energy generation to resume. As a result, for mission-critical systems, putting servers into sleep state and waiting for the renewable energy to resume is not wise, especially for those parallel computing machines with inter-node workload dependency.

Region-II: track wisely. Whenever the load power fluctuates or renewable energy generation varies, load matching is performed as a common practice to handle the power discrepancy [13, 14]. In Region-II, the wind generation oscillates severely. The load power tuning is largely a result of the power supply variation, as shown in Figure 4. However, aggressively matching the load to the supply results in little energy benefits but disturbs the normal server operation and degrades the performance of parallel workload. Therefore, seeking appropriate tracking timing becomes especially important.

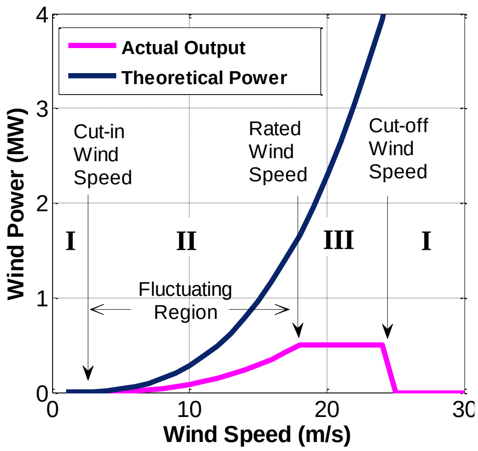


Figure 2: Wind power output characteristics. The cubic relationship between power and wind speed increases the output variability

Region-III: schedule wisely. When the renewable energy is relatively stable, frequent load fluctuation contributes to a number of load tuning operations. In Figure 4 (Phase III), although the data center power has a relatively small dynamic range, frequent variation invokes a large number of back-and-forth load migration operations. Those tuning activities have little contribution to the overall renewable energy utilization but increase network traffic significantly. A well designed job scheduling that mitigates load power fluctuation will help lower the overhead.

We believe a renewable energy powered data center will frequently experience the aforementioned three power management regions throughout its lifetime. To improve overall quality of service, the best design practice should put the above scenarios together and provide a cooperative power management scheme.

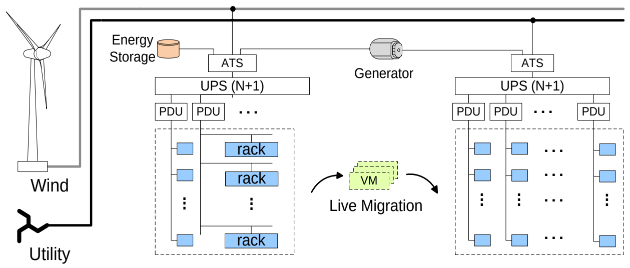


Figure 5: iSwitch load power balancing in wind energy powered server clusters. It does not require increasing the number of servers to handle workload surge

4. An Overview of iSwitch Architecture

In this section we propose *iSwitch*, a holistic data center coordination and optimization scheme that ensures high renewable energy utilization and low operational overhead. As shown in Figure 5, *iSwitch* is designed to provide autonomic load power balancing between conventional utility grid and renewable energy generation.

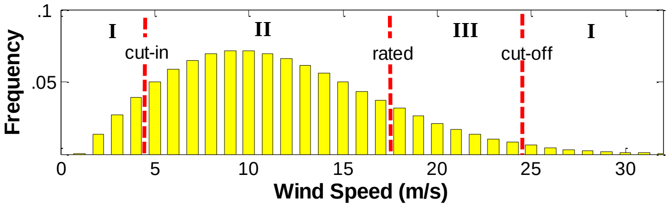


Figure 3: Wind speed variations in most wind farms are typically best described by Rayleigh distribution

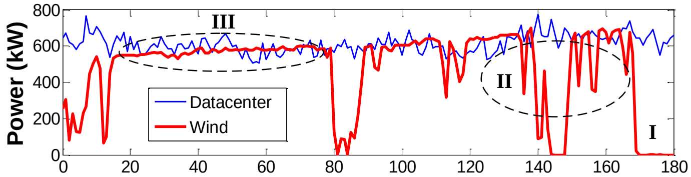


Figure 4: Power variation scenarios in wind energy driven data centers (x-axis: minutes). Region-II and Region-III are two critical regions that require special care

In Figure 5, the on-site renewable power supply provides the data center with clean energy through a separate power line. We choose not to synchronize renewable power to the grid because the fluctuating nature of renewable energy often challenges the grid stability [21]. On the other hand, although one can leverage dual-corded servers to utilize two power supplies simultaneously, it is not energy-efficient when the computing load is low [22]. In this study, *iSwitch* explores computing load migration as an alternative of energy source integration.

4.1. Switching Mechanism

The basic idea behind *iSwitch* is *switching*, or the operation of performing a *switch*. In this study, the *switch* is defined as a load migration event that leads to redistribution of load power between different power supplies.

As an alternative to load power throttling, *iSwitch* intelligently shifts the computing load from one energy source to another to achieve best load power matching. We use virtual machine (VM) live migration to implement *iSwitch* since it is the most convenient way to perform load power shifting in a virtualized computing environment. Existing virtual machine power metering [23] also eases the monitoring and coordination of each individual VM.

Note that in Figure 5, *iSwitch* does not require increasing the number of servers to meet the workload surge. In emergency scenarios, we use backup energy storage to temporarily support the load. According to Fan et al. [24], server clusters can spend more than 80% of the time within 80% of their peak power, and 98% of the time within 90% of their peak power. Therefore, the chance of workload triggered emergency is small. In this study, we assume that the number of renewable energy powered servers is less than 40% of the overall deployed machines since a data center typically consumes about 60% of its actual peak power [24]. In this case, even if the wind power is extremely low, the utility grid can still take over most of the load.

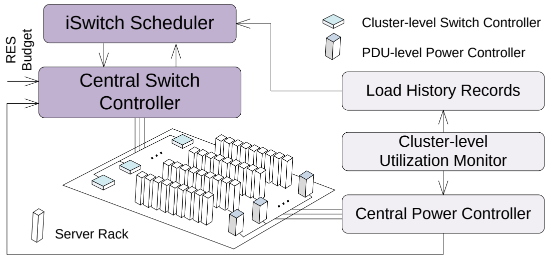

iSwitch Scheduler
RES Budget
Central Switch Controller
Cluster-level Switch Controller
PDU-level Power Controller
Load History Records
Cluster-level Utilization Monitor
Central Power Controller
Server Rack

Figure 6: iSwitch global (facility-level) control

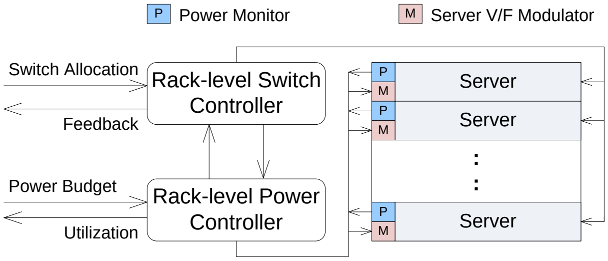


Figure 7: iSwitch local (rack-level) control

4.2. Control Mechanism

To handle the time-varying, intermittent renewable power, *iSwitch* dynamically allocates and de-allocates (i.e., "switches") the renewable energy powered server load. The supply/load variability makes the switch tuning challenging since the control should 1) *globally respect the time-varying renewable budget* and 2) *locally avoid any power failure induced by load fluctuation*. To this end, *iSwitch* uses a hierarchical switching control scheme, which can be easily incorporated into existing hierarchical power management methods such as [25].

Facility level: Figure 6 shows a global view of *iSwitch* control mechanism. The switching operation is controlled by a central switch controller (CSC), which communicates with a central power controller (a typical facility-level data center power controller), a switch scheduler and multiple cluster level switch controllers. CSC performs switch tuning based on the discrepancy between the load power consumption and the RES budget. Whenever needed, switching operations are scheduled by the switch scheduler, which stores profiling information for each server load and optimizes switching using load history records.

Cluster level: The switching allocation is assigned to local computing nodes via cluster-level switching controllers, which are counterparts to PDU-level power controllers. The cluster-level switching controller collects switching outcomes (i.e., the number of switching operations accomplished/failed) of each local computing node and feeds the information to the CSC for switching scheduler updates. The cluster-level controller improves the manageability of dynamic switching and reduces the overhead of CSC communication traffic.

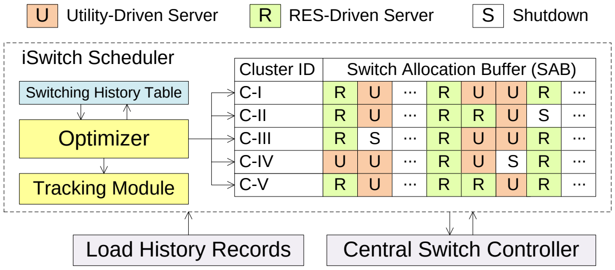
Rack level: As shown in Figure 7, a rack-level switch controller executes power supply switching and sends the execution outcomes to the CSC via a cluster-level switch controller. It also interacts with the rack-level power controller throughout the switching process to avoid any brownout. For example, whenever the power consumption of a server rack reaches its local renewable power budget, the power controller will signal the rack-level switch controller to throttle the switching activities. In addition, the rack-level power controller is able to perform power capping by manipulating the voltage and frequency modulator of the server. This will prevent over-loading if power switching cannot handle the load surge in time.

5. Optimizing Load Tuning Activities

This section proposes the supply/load cooperative optimization scheme of *iSwitch*. Our technique features a lazy tracking scheme on the supply side and a power demand smoothing scheme on the load side. This cooperative optimization is readily supported by existing data center architectures and is orthogonal to other system-level control and workload optimizations.

The switching scheduler is the key architecture for *iSwitch*, as shown in Figure 8. It monitors the power provisioning status (i.e. powered by renewable energy or utility grid) of each server load (i.e., VMs). All the running loads within each cluster are indexed consecutively in a switch allocation buffer (SAB). A switching history table is used to store the switching frequency for each load. An optimizer computes the optimal switching assignment and a tracking module initiates the switching process.

To make load tuning decisions, *iSwitch* scheduler needs profiling information such as server utilization data from the load history table. The central power controller invokes scheduling activities in response to variations in renewable power supply and load power demand. Whenever necessary, the scheduler sends a sequence of load switching commands to the central switch controller for execution.



Cluster ID	Switch Allocation Buffer (SAB)							
C-I	R	U	···	R	U	U	R	···
C-II	R	U	···	R	R	U	S	···
C-III	R	S	···	R	U	U	R	···
C-IV	U	U	···	R	U	S	R	···
C-V	R	U	···	R	R	U	R	···

Figure 8: Architecture of iSwitch scheduler

5.1. Lazy Power Supply Tracking

The first idea of *iSwitch* is to avoid tracking the severely fluctuant renewable power in Region-II (detailed in Section 3). Inside the scheduler, a tracking module manages *iSwitch* power tracking, as shown in Figure 8. We call it lazy tracking because the module only harvests the relatively stable renewable energy generation. Note that *iSwitch* carefully distributes the switching activities across all the loads evenly to avoid local traffic jam.

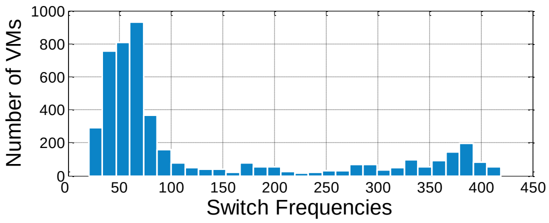


Figure 9: Histogram of switching frequency using round-robin virtual machine selection during load migration

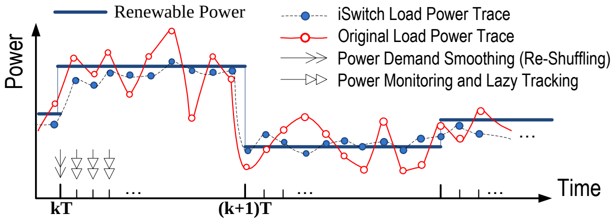

Renewable Power
iSwitch Load Power Trace
Original Load Power Trace
Power Demand Smoothing (Re-Shuffling)
Power Monitoring and Lazy Tracking
Power
Time
kT
(k+1)T
…
…
…
…

Figure 10: Optimization timeline of iSwitch. Lazy tracking is fine-grained and demand smoothing is coarse-grained

Lazy tracking: At each fine-grained interval, when switching is triggered by the CSC, an estimated switching assignment will be sent to the scheduler for calculating the switch operation balance (e.g., estimated assignment minus the baseline). If the switch balance indicates a reduced number of servers to RES connection, the scheduler signals the CSC to schedule the estimated assignment to avoid brownout. On the other hand, if the switching balance suggests an increased number of servers to RES connection (e.g., due to temporally decreased load or increased supply), the scheduler will signal CSC only if the balance is larger than a preset threshold (e.g., 10% of the baseline). In this case, we ignore the high-frequency switching surge which brings little benefit on renewable energy utilization but leads to excessive unnecessary load migration.

LRU distribution: Within each cluster, *iSwitch* allocates switch operations with least recently used (LRU) method, which avoids aggressively tuning a small set of computing nodes. Note that a naive switching allocation can result in unbalanced switching allocation. In Figure 9 we show the switching distribution as a result of round-robin scheduling. The average switching frequency is 200 times per day per VM. A small group of VMs receives up to 400 times per day. As a result, some racks may incur more performance penalty due to high communication traffic.

To implement LRU, *iSwitch* uses the switch frequency record stored in the switching history table. The operation of *iSwitch* scheduler relies on the load history record of the previous control period. This record can be implemented using a round-robin database (circular buffer) with constant storage occupation over time. Such round-robin database is easy to reconfigure and manage in most Linux servers with RRDtool [26], which is an industry standard, high performance data logging system.

5.2. Load Power Demand Smoothing

Optimizing the supply-side fluctuation alone cannot achieve significant overhead mitigation (detailed in Section 7.1). To this end, *iSwitch* leverages the heterogeneity of server clusters to minimize load fluctuation-induced overhead in power management of region III (Section 3).

Figure 10 illustrates the switch management timeline of *iSwitch*. The controller re-shuffles the renewable energy powered servers at a coarse-grained time interval *R* (e.g., 15 minutes as the default value in our experiments). During each re-shuffling interval, the average load utilization is recorded in a fine-grained time interval (e.g., 1 minute) and is used to predict the load for the next period. Upon rescheduling, the optimizer in the *iSwitch* scheduler updates the baseline switch operations of each server cluster in SAB with the goal of mitigating the likelihood of severe load power fluctuation in the next control period. Each switch tuning invoked by CSC will be assigned based on the updated SAB.

At the beginning of each control period, *iSwitch* recalculates the optimal switch assignment. To simplify the problem, we assume the data center servers are logically divided into *c* clusters and the load is balanced within each cluster (i.e., almost homogeneous server utilization).

Let $U_i = [u_{i1} \quad u_{i2} \quad \cdots \quad u_{ic}]$ denotes the average utilization of each cluster at time stamp *i*. The utilization history record for the previous control period that consists of *m* time stamps is:

$$\boldsymbol{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1c} \\ u_{21} & u_{22} & \cdots & u_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mc} \end{bmatrix} \tag{2}$$

Assuming a total number of N virtual machines is to be connected to the renewable power supply in the next control period. The migration decision for the next control period is $\boldsymbol{S} = [s_1 \quad s_2 \quad \cdots \quad s_k \quad \cdots \quad s_c]$, where s_k is the number of VMs selected to be tuned for cluster *k*. To reduce unnecessary load tuning in the future, we want the aggregate power consumption of the selected VMs to have small oscillations in the next control period. In other words, the standard deviation of the aggregate utilization should be minimized. The aggregate utilization is given by:

$$\left[a_{ij}\right]_{m\times 1} = \boldsymbol{U} \times \boldsymbol{S}^T \tag{3}$$

The standard deviation of the expected utilization in next control period can be calculated as:

$$\sigma = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(a_i - \mu)^2} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}a_i^{\,2} - (\frac{1}{m}\sum_{i=1}^{m}a_i)^2}\ ,$$

$$and \qquad a_i = \sum_{k=1}^{c} u_{ik} s_k \tag{4}$$

In Equation-4, a_i is the aggregate utilization of renewable energy powered load and μ is the mean utilization during the past control period *R*. The re-shuffling problem is therefore formulated as:

$$\textit{Objective}: \quad \min\{\frac{1}{m}\sum_{i=1}^{m} a_i^2 - (\frac{1}{m}\sum_{i=1}^{m} a_i)^2\} \qquad (5)$$

$$\textit{Constraints}: \sum s_k \le N \qquad (6)$$

We solve the above non-linear minimization problem with simulated annealing (SA). Given the utilization history records, our SA solver is capable of finding the desired global extreme very fast. Note that the renewable power supply fluctuation typically occurs on a coarse-grained time interval (several minutes). As a result, the execution time (several seconds in our experiments) of SA solver does not affect the optimization effectiveness. At the beginning of the re-shuffling period, the switching operations are assigned in proportion to the number of servers in the cluster. During the computation, the SA solver iteratively generates a stochastic perturbation for the switching assignment and checks whether or not the optimum solution is reached.

6. Evaluation Methodologies

We evaluate our design with trace-driven simulation. We have developed a framework that simulates dynamic load tuning and hierarchical power control in renewable energy powered data centers. For each scheduled job requests, we calculate its contribution to the overall data center power consumption based on the number of nodes requested and the job's specific resource utilization statistics. Our framework takes realistic wind energy generation traces as supply-side input.

6.1. Data Center Configurations

We assume a raised floor data center consisting of 4,800 servers and the servers are organized as twelve rows with each row powered by a 100KW PDU. There are ten 40U server racks in each row and the server we modeled resembles an HP ProLiant DL 360 G6. The peak and idle power of the modeled server are 186W and 62W respectively. We convert the server utilization traces to its power consumption using the published SPECPower results [27], which have been widely used for data center power evaluation [7, 8]. Since the SPECPower results only reflect the server power at intervals of 10% utilization, we use linear interpolation to approximate the power across different load levels.

We evaluate data centers with both homogeneous and heterogeneous load variations. The homogeneous workload configuration assumes that all the servers are running the same workload and have similar utilization levels. As shown in Table 1, we generate the homogeneous utilization traces from the raw data provided by the Internet Traffic Archive [28]. We convert the request rate (requests per minute) into server utilization by investigating the maximum request rate (corresponding to 100% loading) and minimum request rate (corresponding to 0% loading). The server utilization traces we generated represent a one-week server load variation including idle period, peak hours and daily surge.

Our heterogeneous data center configuration is based on an academic high-performance computing (HPC) center, which hosts more than 600 servers. The HPC center has five major clusters (C-I~C-V) with different service targets and loads, as detailed in Table 2. Those clusters have 20 to 400 computing nodes and their average utilization ranges from 25% to more than 90%. All the clusters are managed with RRDtool [26], which enables autonomic data logging and trace generation. Since we have limited access to industrial data center traces, we collect real-world workload logs from a well-established online repository [29]. The workload logs provide information such as job arrival times, start time, completion time, size in number of computing nodes, etc. We choose the "cleaned" version of each trace log [30]. These traces have been already scrubbed to remove workload flurries and other anomalous data that could skew the performance evaluation [30].

Table 3 summarizes our evaluated workload trace combinations. We build various workload mixes to mimic today's data centers that are composed of many small co-located clusters. Each workload set in Table 3 consists of five traces [29] which run on the aforementioned five clusters (C-I to C-V) respectively. To form representative workload sets, we characterize workload traces based on their average job size and runtime. In Table 3, *Mix-High* includes traces that have larger job size (resulting in >30% average data center utilization) and *Mix-Low* contains traces that have small job size (resulting in <10% utilization). On the other hand, *Mix-Stable* consists of five traces that feature relatively longer job runtime and *Mix-Bursty* consists of traces that have very short job runtime.

Trace	**Description**	**Avg. Loading**	**Load Level**
Calgary	University Web Server	2.8%	Very low
U of S	University Web Server	7.5%	Low
NASA	Kennedy Space Center Server	27.8%	Moderate
Clark	Clark WWW Server	33.4%	High
UCB	UC Berkeley IP Server	43.2%	Busy

Table 1: Traces of homogeneous server utilization [28]

	Cluster ID	**% of Overall Deployed Servers**	**Avg. Loading**
HPC	C-I	5%	97%
	C-II	63%	60%
	C-III	17%	57%
	C-IV	3%	54%
	C-V	12%	25%

Table 2: Configuration of heterogeneous clusters

Workload Set	Description	Workload Trace Combination
Mix-High	High utilization	*“HPC2N” + ”LANL CM5” + ”LPC EGEE” + “SDSC BLUE” + “LLNL Thunder”*
Mix-Low	Low utilization	*“DAS- fs0” + “DAS2-fs1” + “DAS2-fs2” + “DAS2-fs3” + “DAS2-fs4”*
Mix-Stable	Stable power demand	*“HPC2N” + “KTH SP2” + “LANL CM5” + “DAS2-fs0” + “SDSC BLUE”*
Mix-Bursty	Bursty power demand	*“DAS2-fs2” + “DAS2-fs3” + “DAS2-fs4” + ”LPC EGEE” + “OSC Cluster”*
Mix-Rand	Random combination	*“LLNL Thunder” + “OSC Cluster” + ”LPC EGEE” + ”LANL CM5” + “KTH SP2”*
Dept-HPC	Traces collected from departmental high-performance computing center	

Table 3: The evaluated heterogeneous datacenter workload sets. Each workload set consists of five parallel workload traces [29] which are fed to clusters C-I~C-V shown in Table 2

	Trace Abbr.	Wind Energy Potential	Locations (Station ID)	Capacity Factor (CF)	Power Density
I	W1	Low	California (9250)	15%	195 W/m^2
	W2	Medium	Arizona (6107)	25%	338 W/m^2
	W3	Typical	Colorado (10563)	35%	581 W/m^2
	W4	High	Texas (1360)	45%	708 W/m^2
II	LVS	Low variation trace	Wyoming (15895)	50%	1021 W/m^2
	HVS	High variation trace	Utah (11967)	21%	607 W/m^2

Table 4: The evaluated wind power supply traces [31]. Group-I highlights different wind power generation potentials. Group-II focuses on variation intensity of the supply and is used for characterization purposes

Abbr.	Design Philosophy	Tracking	Stored Energy	Load Deferment
Utility	Fully relies on utility grid	No	No	No
Battery	Relies on battery to provide reliable renewable power	No	Yes	No
Green	Focuses on sustainability, 100% renewable energy powered	Yes	No	Yes
Tracking	Maximizes energy utilization with aggressive supply tracking	Yes	Yes	No
iSwitch	Aims at achieving high sustainability, low overhead and latency	Yes	Yes	No

Table 5: The evaluated power management schemes

6.2. Renewable Energy Supply Traces

We use wind power data traces from The Wind Integration Datasets [31] of the National Renewable Energy Laboratory. These wind generation datasets are time-series data of wind power output derived from commercially prevalent wind turbines characteristics and the manufacturer’s rated wind power curve. We carefully selected two groups of traces across different geographic locations and their characteristics are listed in Table 4.

In Table 4, capacity factor (CF) is the ratio of the actual wind turbine output to the theoretical peak generations. Since we are interested in the supply variation and intermittency, we have selected a group of traces with various CF values (i.e., Group-I). While a typical capacity factor of wind turbine is 30% [15], higher capacity factor usually represents better wind energy potential, small power variation and less generation stoppage.

The total installed wind turbine capacity in this study equals to the nameplate power of the studied data center. The actual power budget is therefore only affected by the capacity factor. Note that we also evaluate power supply traces of two extreme scenarios (i.e., Group-II): one has very smooth and stable generation and the other has high output fluctuation rate. All the other renewable supply traces can be seen as a combination of the two basic traces plus the intermittently unavailable periods.

7. Experimental Results

This section quantifies the performance and efficiency of *iSwitch* on a wide range of workload configurations. We first characterize the impact of supply/load power variability on data center load matching using homogeneous workload traces. We then compare the performance overhead and energy utilization of *iSwitch* to some state-of-the-art approaches. In Table 5, *Utility* and *Battery* are two conventional schemes which do not involve supply-driven load matching. *Green* is the most sustainable design. *Tracking* represents emerging design approaches [13, 14] which leverage load adaptation to actively track every joule of renewable energy generation.

7.1. Impact of Power Variability

To understand how power variation affects load matching behavior, we monitored the load migration frequency of each VM across different server utilization levels and renewable power supply scenarios. Figure 11 shows the average switching frequency for all VMs for one week’s duration. Figure 12 characterizes the standard deviation of the migration activities of a single VM. In Figures 11 and 12, *Idle*, *Typical* and *Busy* are data center traces that have average utilization levels of 0%, 30% and 60%, respectively. We generate these three synthetic data center traces to mimic low load variation data centers.

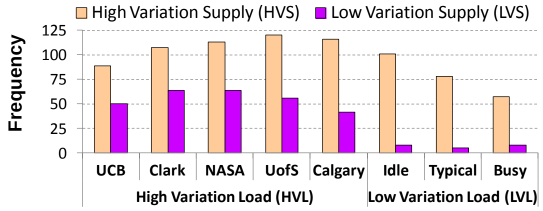


Figure 11: Average load switching frequency

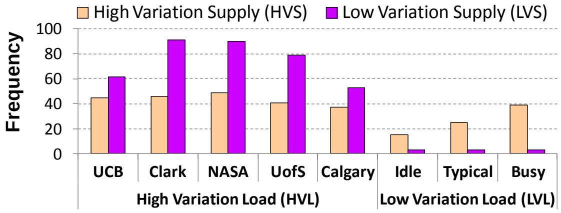


Figure 12: Standard deviation of the switching operation

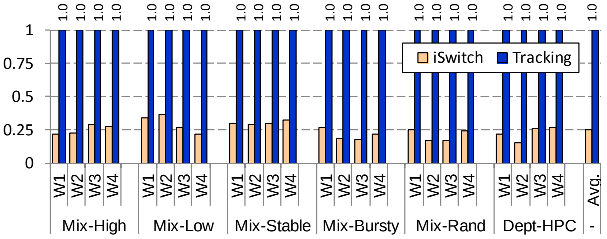


Figure 13: The average network traffic (all the results are normalized to *Tracking*)

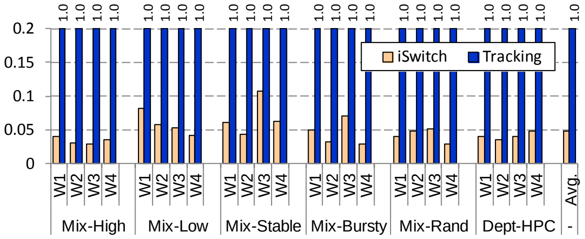


Figure 14: The peak network traffic (all the results are normalized to *Tracking*)

High variation load (HVL): When the supply variation is high (i.e. *HVS*), heavy switch tuning across a large group of data center servers is common. Therefore, the average switch frequency is high and the standard deviation is not very large. For low supply variation (i.e. *LVS*), however, the standard deviation increases by 66% since the switching triggered by load oscillations typically stresses a relatively small group of server loads.

Low variation load (LVL): Since the load variation is less severe, the total switching activities are reduced in both cases (i.e. *HVS* or *LVS*) and the average switching frequency is small. For example, *Typical* has similar loading compared to *NASA*. However, the former reduces average switching frequency by 31% when the supply variation is high (i.e. *HVS*) and by 90% when the supply variation is low (i.e. *LVS*). In Figures 11 and 12, a combination of *LVL* and *LVS* manifests the lowest control effort since the mean value and standard deviation of per VM switching are both small.

To summarize, choosing a subset of server load that has lower total power variation can significantly reduce the load switching demand, especially when the supply variation is low. When the supply variation is high, simply dampening the load power variation has limited impact and in this case switch capping can be used to avoid unnecessary tuning.

7.2. Operational Overhead

Frequent load matching activities result in operational overhead, which is our primary concern in the design of renewable energy powered computing systems. In a virtualized environment, *iSwitch* could effectively reduce the VM migration rate and help to save data center network bandwidth significantly. The data migration traffic is calculated at rack-level. Each VM live migration transfers approximately the size of the VM's memory between hosts [32]. We assume a VM memory size of 1.7GB in our calculation, which is the default memory size of Amazon EC2 standard instance.

Figure 13 shows the average communication traffic across various workload configurations and wind power supply levels. All the results are normalized to *Tracking*. We do not show the results of *Green* because it has the same power tracking frequency as *Tracking*. As can be seen, on average, *iSwitch* could reduce 75% of the rack-level traffic and therefore significantly releases the network bandwidth burden. The results are even more impressive for peak traffic hours when the renewable energy fluctuates severely. In Figure 14 we calculate the communication traffic during the top 1% high-traffic hours. Because *iSwitch* puts a limit on the power tracking activities during fluctuant supply period, it shows only 5% network traffic compared with *Tracking*.

7.3. Latency per Hour

Another advantage of *iSwitch* is that it reduces the migration frequency of each VM instance and thereby improves the job turnaround time. Due to the intermittency of renewable generation, stand-alone systems such as *Green* experience long waiting time (about tens of minutes as observed in our experiments). These systems typically leverage deferrable load [33] to meet the power budget or simply perform load shedding [34] to avoid brownout. Even for utility-connected systems such as *iSwitch* and *Tracking*, latency exists due to the data migration time. For example, the time needed for a 1GB VM migration takes about 20 seconds to complete [35].

In this study, we use latency per hour (LPH) to evaluate the performance overhead. For example, a LPH of 10 means each individual VM instance experiences 10 seconds waiting time per hour on average. Figure 15 shows the average LPH across the entire data center servers. The average LPH of *iSwitch* is about 30 seconds per hour while the average LPH of *Tracking* reaches 126 seconds per hour. The average LPH of *Green*, however, is about 1500 seconds per hour – 50 times that of *iSwitch*. Therefore, waiting for renewable energy to resume (e.g. *Green*) should be the last resort.

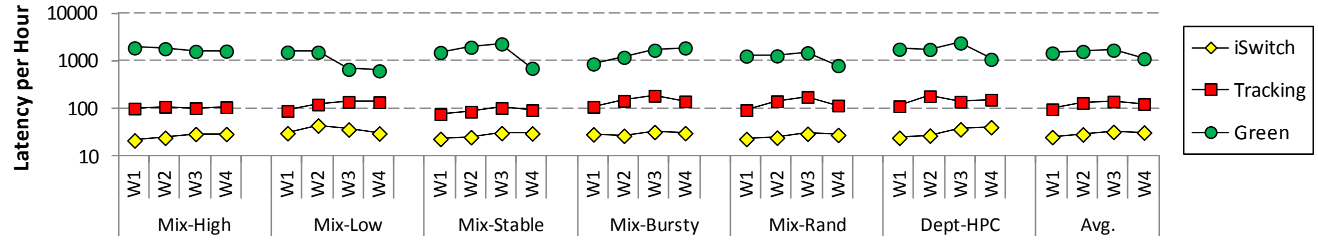


Figure 15: The average latency of job requests

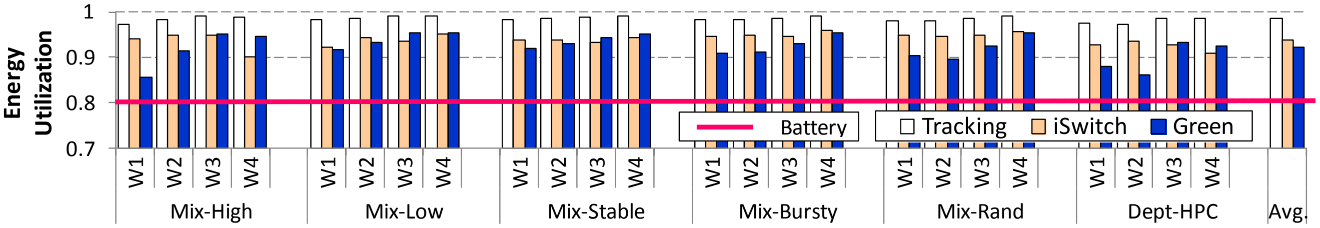


Figure 16: The normalized overall wind energy utilization

7.4. Renewable Energy Utilization

We evaluate the renewable energy utilization (REU) of data centers with different wind power provisioning capacities and workload behaviors. The REU is defined as the amount of wind energy that is actually utilized by the load divided by the total amount of wind energy generation. A higher REU indicates better supply/demand coordination, which reduces on-site energy storage capacity, improves return-on-investment (ROI) and data center sustainability, and eases the initial infrastructure planning.

While *iSwitch* uses a lazy power tracking scheme, it does not sacrifice energy utilization significantly. As shown in Figure 16, *iSwitch* can achieve an average renewable energy utilization of 94% – higher than *Green* (92%) but lower than *Tracking* (98%). The reason why *Tracking* outperforms *iSwitch* on energy utilization is that *Tracking* tracks every joule of wind energy generation aggressively. Note that a 4% decrease in energy utilization does not mean that *iSwitch* is less preferable in our study; *iSwitch* significantly reduces network traffic and improves the performance by 4X. In contrast to *Tracking*, *iSwitch* trades off energy utilization for better job turnaround time.

7.5. Optimization Effectiveness

Improperly setting the *iSwitch* re-shuffling intervals leads to degraded optimization effectiveness. To understand the tradeoff, we characterize the design space by varying the load re-shuffling intervals. In the following discussion, *S-x* means *iSwitch* with a re-shuffling interval of *x* minutes.

We analyze the average network traffic under various re-shuffling intervals. In Figure 17-a, all the results are normalized to *S-15*. It shows that increasing the re-shuffling interval could mitigate the overhead and reduce network traffic. For example, *S-120* manifests 35% traffic reduction compared with *S-15*. However, an extended re-shuffling interval could also degrade *iSwitch* energy utilization due to the decreased adaptivity. We evaluate the impact of long re-shuffling periods on direct renewable energy utilization (DREU), as shown in Figure 17-b. Here, direct renewable energy utilization means the renewable energy directly utilized by the system without passing through batteries. Compared to *S-15*, *S-120* yields about 24% DREU degradation (which means increased battery capacity is required to store the remaining generation).

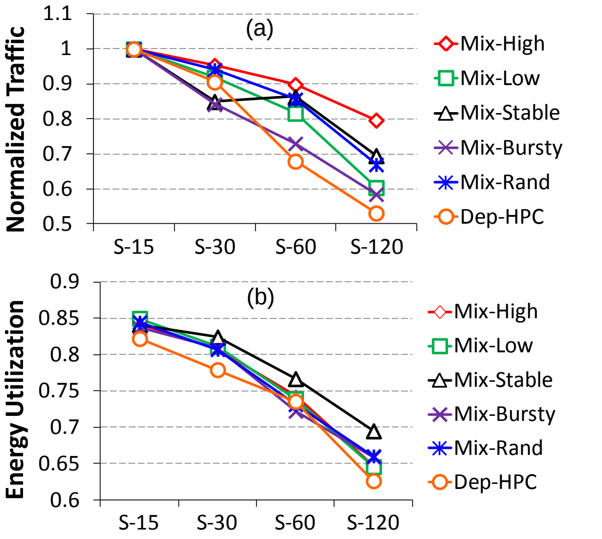


Figure 17: iSwitch with different control intervals. The results show all six workload trace sets. The wind energy trace used is *W3* which has typical wind energy potential

To understand DREU degradation, we show a fraction of data center power consumption trace controlled by *S-120*, which uses a two-hour load history record as prediction input. In Figure 18, *iSwitch* does not react to gradual renewable power supply increase. We recommend a control period of 15~30 minutes. In this case, *iSwitch* could reduce 75% average load tuning while still maintaining more than 80% direct wind energy utilization (94% overall wind energy utilization if combined with battery).

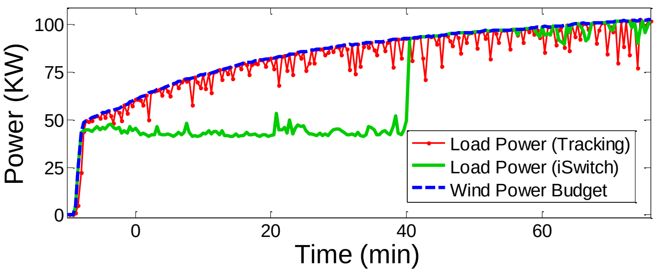


Figure 18: The impact of long re-shuffling interval of iSwitch

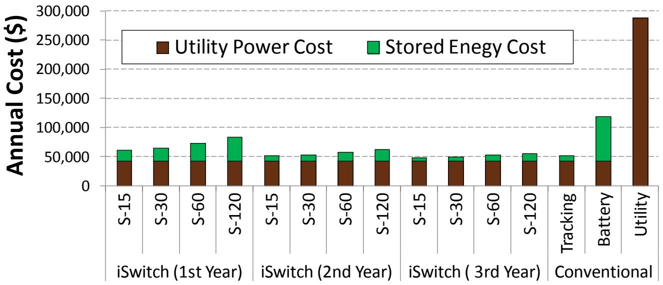


Figure 19: Annual operating cost of various design scenarios

In sum, the length of *iSwitch* re-shuffling interval affects *iSwitch* load matching in two ways. First, short re-shuffling intervals limit the load fluctuation mitigation because a short load history record gives less accurate prediction. Second, a shorter time interval means more frequent re-shuffling and therefore magnified overall control overhead. This control overhead arises because the SA solver in *iSwitch* may request additional switch tuning to mitigate the total variation. For example, instead of directly switching 100 virtual machines from utility to RES, the scheduler may first disconnect 20 high-variation wind-powered VM instances and then connect 120 low-variation VM instances to the wind turbine.

7.6. Total Cost of Ownership

Because *iSwitch* is capable of utilizing the renewable power directly, we save a large amount of battery capacities. Otherwise, we have to use a large-scale battery to store the unused excess generation, which is not economic and environment-friendly.

In Figure 19 we show the projected annual operating cost of designing a wind energy-powered small-scale data center (i.e. 890KW server deployment). The average retail price of electricity in industrial sector is about $0.07/KW. The estimated energy storage cost is $300/KW for lead-acid batteries [36]. A lower direct renewable energy utilization rate leads to increased battery capacity demand. In Figure 19, the operating cost of *S-120* in the first year is 35% of a utility powered data center (*Utility*). After three years, the average annual operating cost is only 20% of *Utility*. The implementation cost is amortized by the renewable energy in the following deployment duration.

8. Related Work

While power-efficient computer architecture is a well studied area in both industry and academia, designing renewable energy powered computing systems gained its popularity only recently [13, 14, 37, 38, 39]. In this section we highlight a number of representative works that strive to improve IT efficiency and sustainability in different aspects.

Data center power management: Substantial research has been done on optimizing power efficiency of data centers. At the server level, the most widely used tuning knobs are dynamic voltage and frequency scaling (DVFS), power capping and power state switching [7, 40]. At data center level, various mechanisms can be found in recent proposals. For example, some major techniques include virtual machine resource monitoring and capping [23], job scheduling with queueing theory [11], dynamic data channel width adaptation [41], and intelligent server consolidation for co-optimization of cooling and idle power [9, 10]. Although all those works can be leveraged to track the time-varying renewable power supply, none of them is aware of the supply variation attributes. As a result, they incur significant power management overhead but gain very limited energy efficiency return.

Emerging green computing techniques: Power intermittency is the most significant challenge in a renewable energy powered computing system. To avoid brownout, recent studies propose load deferment [33] and load shedding [34] to match the demand to the supply. Both approaches are not suitable for data centers which have strict performance requirement specified in the service level agreement (SLA). Although instantaneous performance cannot be guaranteed, one can still use load power adaptation to improve the overall renewable energy utilization and optimize the workload performance with additional power budget [13, 14]. Additionally, one can further leverage various renewable energy integration points [39], energy storage elements [42], and distributed UPS system [43] to improve renewable energy utilization. In contrast to existing work, this paper explores the benefits of putting utility power, energy storage, and load migration coordination together. The key novelty of our design is that we propose a supply/load cooperative optimization that significantly reduces existing renewable power management overhead while still maintaining desirable renewable energy utilization.

9. Conclusions

Environmental and energy price concerns have become key drivers in the market for sustainable computing. The advance of renewable energy technologies and continuously decreasing renewable power cost have made renewable energy driven data centers a proven alternative to conventional utility-dependent data centers and the market is rapidly growing.

Matching the variable load power consumption to the intermittent power supply appropriately is the crux of designing a renewable energy powered data center. Conventional workload-driven power management has less adaptivity to the power supply variation while existing power-driven system control schemes are not aware that they are experiencing unnecessary and redundant tuning activities. As a result, we either miss the opportunity of utilizing precious renewable energy or incur significant load tuning overhead and performance degradation.

In this study, we propose *iSwitch*, a renewable energy-driven power tuning scheme that addresses a two-fold challenge: the first is to manage intermittent renewable power without sacrificing performance due to power throttling; the second is to ensure high energy utilization with minimized load matching activities. To the best of our knowledge, this is the first paper that digs into renewable power variation characteristics and introduces a supply/load cooperative optimization scheme that minimizes power management overhead. Compared to existing designs, our technique could reduce job waiting time by 80%, mitigate average network traffic by 75%, and rush hour traffic by 95%. Moreover, we can still maintain 96% of the energy efficiency and incur only minor cost increase. This paper shows that we can efficiently handle intermittent renewable power in a lightweight and easy-to-implement manner. We expect that our design will help data center designers take a big step forward in the green computing era.

Acknowledgements

This work is supported in part by NSF grant CSR-1117261, by Microsoft Research Safe and Scalable Multi-core Computing Award, and by NSFC grant 61128004. Chao Li is also supported by a University of Florida Graduate Fellowship.

References

[1] J. Koomey, Worldwide Electricity Used in Data Centers, Environmental Research Letters, 2008

[2] Greenpeace: Cloud Contributes to Climate Change, http://www.datacenter knowledge.com/archives/2010/03/30

[3] http://www.aiso.net/technology-network-sun.html

[4] http://www.solarenergyhost.com/

[5] http://www.greenhousedata.com/greening-the-data-center/

[6] http://eshop.macsales.com/green/wind.html

[7] D. Meisner, B. Gold and T. Wenisch, PowerNap: Eliminating Server Idle Power, ASPLOS, 2009

[8] S. Pelley, D. Meisner, P. Zandevakili, T. Wenisch and J. Underwood, Power Routing: Dynamic Power Provisioning in the Data Center, ASPLOS, 2010

[9] F. Ahmad and T.N. Vijaykumar, Joint Optimization of Idle and Cooling Power in Data Centers While Maintaining Response Time, ASPLOS, 2010

[10] E. Pakbaznia and M. Pedram, Minimizing Data Center Cooling and Server Power Costs, ISLPED, 2009

[11] A. Gandhi, M. Harchol-Balter, R. Das and C. Lefurgy, Optimal Power Allocation in Server Farms, SIGMETRICS, 2009

[12] British Wind Energy Association: http://www.bwea.com/ref/stop.html

[13] C. Li, W. Zhang, C. Cho and T. Li, SolarCore: Solar Energy Driven Multi-core Architecture Power Management, HPCA, 2011

[14] N. Sharma, S. Barker, D. Irwin, and P. Shenoy, Blink: Managing Server Clusters on Intermittent Power, ASPLOS, 2011

[15] M. Patel, Wind and Solar Power Systems, CRC Press, 1999

[16] Renewable Energy Technology Characterizations, Technical Report of National Renewable Energy Laboratory, 1997

[17] Renewable Energy and the Future of the Datacenter, http://arstechnica.com/business/news/2008/05/renewable-energy-and -the-future-of-the-datacenter.ars

[18] IEEE Guide for Array and Battery Sizing in Stand-Alone Photovoltaic (PV) Systems, IEEE Standard 1562-2007, 2008

[19] A. Jossen, J. Garche and D. Sauer, Operation Conditions of Batteries in PV Applications, Solar Energy, 2004

[20] Green Power Network: Net Metering, http://apps3.eere.energy.gov/ greenpower/markets/netmetering.shtml, 2011

[21] J. Peҫas Lopesa,N. Hatziargyrioub, J. Mutalec, P. Djapicc, and N. Jenkinsc, Integrating Distributed Generation into Electric Power Systems: A Review of Drivers, Challenges and Opportunities, Electric Power Systems Research, Vol. 77, Issue 9, 2007

[22] M. Ton and B. Fortenbery, DC Power for Improved Data Center Efficiency, Technical Report, Lawrence Berkeley National Laboratory, 2008

[23] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. Bhattacharya, Virtual Machine Power Metering and Provisioning, ACM Symposium on Cloud Computing, 2010

[24] X. Fan, W. Weber and L. Barroso, Power Provisioning for a Warehouse-sized Computer, ISCA, 2007

[25] X. Wang, M. Chen, C. Lefurgy, and T. Keller, SHIP: Scalable Hierarchical Power Control for Large-Scale Data Centers, PACT, 2009

[26] RRDtool, http://www.mrtg.org/rrdtool/

[27] SPECPower Results, http://www.spec.org/power_ssj2008

[28] The Internet Traffic Archive, http://ita.ee.lbl.gov/

[29] Logs of Real Parallel Workloads from Production Systems, http://www.cs.huji.ac.il/labs/parallel/workload/logs.html

[30] D. G. Feitelson and D. Tsafrir, Workload Sanitation for Performance Evaluation, ISPASS, 2006.

[31] Western Wind and Solar Integration Study, http://www.nrel.gov/ wind/systemsintegration/wwsis.html

[32] V. Soundararajan and J. Anderson, The Impact of Management Operations on the Virtualized Datacenter, ISCA, 2010

[33] A. Papavasiliou and S. Oren Supplying Renewable Energy to Deferrable Loads: Algorithms and Economic Analysis, IEEE Power and Energy Society General Meeting, 2010

[34] H. Xu, U. Topcu, S. Low, C. Clarke, and K. Chandy, Load-Shedding Probabilities with Hybrid Renewable Power Generation and Energy Storage, Annual Allerton Conference on Communication, Control, and Computing, 2010

[35] M. Zhao and R. Figueiredo, Experimental Study of Virtual Machine Migration in Support of Reservation of Cluster Resources, in International Workshop on Virtualization Technology in Distributed Computing, 2007

[36] J. Aabakken, Power Technologies Energy Data Book Forth Edition, National Renewable Energy Laboratory, 2006

[37] K. Le, O. Bilgir, R. Bianchini, M. Martonosi, and T. Nguyen, Capping the Brown Energy Consumption of Internet Services at Low Cost, IGCC, 2010

[38] Z. Liu, M. Lin, A. Wierman, S. Low and L. Andrew, Greening Geographical Load Balancing, SIGMETRICS, 2011

[39] N. Deng. and C. Stewart, Concentrating Renewable Energy in Grid-Tied Datacenters , International Symposium on Sustainable Systems and Technology, 2011

[40] S. Kaxiras and M. Martonosi, Computer Architecture Techniques for Power-Efficiency, Morgan & Claypool, 2008

[41] D. Abts, M. Marty, P. Wells, P. Klausler and H. Liu, Energy-Proportional Data Center Networks, ISCA, 2010

[42] S. Govindan, A. Sivasubramaniam and B. Urgaonkar , Benefits and Limitations of Tapping into Stored Energy for Datacenters, ISCA, 2011

[43] V. Kontorinis, L. Zhang, B. Aksanli, J. Sampson, H. Homayoun, E. Pettis, D. Tullsen, and T. Rosing, Managing Distributed UPS Energy for Effective Power Capping in Data Centers, ISCA, 2012