GreenHetero: Adaptive Power Allocation for Heterogeneous Green Datacenters

Haoran Cai*†, Qiang Cao*✉, Hong Jiang§, Qiang Wang*
*Wuhan National Laboratory for Optoelectronics, Key Laboratory of Information Storage System of Ministry of Education,
School of Computer Science and Technology, Huazhong University of Science and Technology
§Department of Computer Science and Engineering, University of Texas at Arlington
†Algorithm and Technology Development Department, Carrier BG, Huawei Technologies Co., Ltd
✉Corresponding Author: caoqiang@hust.edu.cn

***Abstract*—In recent years, the design of green datacenters and their enabling technologies, including renewable power managements, have gained a lot of attraction in both industry and academia. However, the maintenance and upgrade of the underlying server system over time (e.g., server replacement due to failures, capacity increases, or migrations), which make datacenters increasingly more heterogeneous in their key processing components (e.g., capacity and variety of processors, memory and storage devices), present a great challenge to optimal allocation of renewable power supply. In other words, the current heterogeneity-unaware power allocation policies have failed to achieve optimal performance given a limited and time varying renewable power supply.**

In this paper, we propose a dynamic power allocation framework called GreenHetero, which enables adaptive power allocation among heterogeneous servers in green datacenters to achieve the optimal performance when the renewable power varies. Specifically, the GreenHetero scheduler dynamically maintains and updates a performance-power database for each server configuration and workload type through lightweight profiling method. Based on the database and power prediction, the scheduler leverages a well-designed solver to determine the optimal power allocation ratio among heterogeneous servers at runtime. Finally, the power enforcer is used to implement the power source selections and the power allocation decisions. We build an experimental prototype to evaluate GreenHetero. The evaluation shows that our solution can improve the average performance by 1.2x-2.2x and the renewable power utilization by up to 2.7x under tens of representative datacenter workloads compared with the heterogeneity-unaware baseline scheduler.

I. Introduction

Modern datacenters are consuming increasing amounts of electricity. According to a 2015 NRDC report [4], datacenter electricity consumption is projected to reach roughly 140 billion kilowatt-hours annually by 2020. This is the equivalent annual output of 50 power plants, costing U.S. businesses $13 billion annually in electricity bills. Such huge IT energy consumption not only increases the total cost of ownership (TCO) but also leaves profound impact on the environment. According to another report, the annual CO_2 emissions of computing systems will reach 1.54 metric gigatons within eight years, which could make IT companies the biggest greenhouse gas emitters by 2020 [16]. To this end, many cloud service providers, such as Apple [36] and McGraw-Hill [38], have built their datacenter power systems using on-site generation of renewable energy from sources such as wind and solar. Such green design can significantly reduce the environmental footprint and TCO of datacenters.

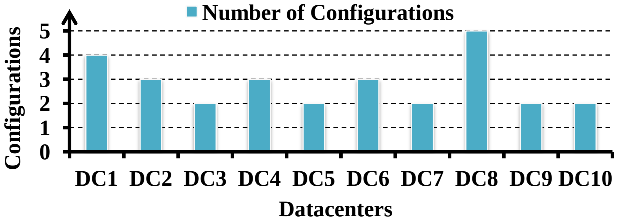


Fig. 1: Numbers of server configurations in ten different Google datacenters [22].

In many prior studies, researchers have built green power systems [21], [11] and proposed relevant power management schemes [35], [10], [9], [17], [32]. However, these studies are, by and large, based on the assumption that the computing environments are homogeneous in modern datacenters. In practice, on contrary, these datacenters are typically composed of replaceable commodity components [23]. To pursue a higher performance target, datacenter managers will gradually deploy new generations of hardware while parts of the older-generation hardware remain and continue to operate. As a result, a typical datacenter evolves to be a mixture of server platforms and indeed becomes a *heterogeneous datacenter*. Figure 1 presents the server diversity found in ten randomly selected Google datacenters in operation in terms of the number of different configurations in each of these datacenters [22]. According to this figure, these datacenters are deployed a number of microarchitectural configurations ranging from 2 to 5, including both Intel and AMD servers from several generations. As the requirements of big data processing and AI computation are growing exponentially, the owners of datacenters also deploy high-performance computing (HPC) platforms (e.g., GPUs) and domain-specific accelerators (e.g. TPU and FPGA-based). Therefore, heterogeneous servers have been prevalently deployed in modern datacenters.

However, existing studies on provisioning of renewable energy largely ignore such configuration heterogeneity in green datacenters. Considering the time-varying and intermittent nature of renewable power supply, an important research question is how to allocate appropriate amount of power to each heterogeneous server, especially when the power supply is insufficient. Specifically, the uniform power allocation strategy used in 'homogeneous' datacenters by default will distribute unbalanced power to the actual heterogeneous servers, leading to power wastage and performance degradation. Compounding the challenges posed by the time-varying and intermittent nature of renewable power supply, the various workload types and server configurations in datacenters make it ever more difficult for power management in a green datacenter to determine an optimal power allocation scheme to achieve the best performance.

In this work, we propose GreenHetero, a dynamic power allocation framework that aims at exploring the optimal power allocation schemes and improving overall performance in heterogeneous green datacenters. Specifically, we design a new adaptive scheduler to determine the appropriate power sources (e.g., renewable power, battery energy and grid power) and power allocation schemes. First, based on renewable power and server rack power predictions, the scheduler divides the solution into three cases and chooses the power sources accordingly. When the renewable power supply can fully satisfy the power demand of server racks, it will independently sustain the power load and the surplus renewable power will be charged into energy storage devices. When the renewable power supply is insufficient or shows great fluctuation, the batteries, strategically charged either by renewable source or grid source, discharge to make up for the power shortage. When the renewable power is unavailable, the batteries individually supply the load power. Second, after power source selections have been determined, the scheduler will choose a specific power allocation management. To tackle the server and workload heterogeneity challenge, we introduce a solver to find the optimal power allocation ratio based on a well-constructed profiling database.

This paper makes the following contributions:

- We introduce a new metric called Effective Power Utilization (EPU) and analyze the impact of different renewable power allocation schemes on performance and EPU for heterogeneous datacenter servers. We find that an appropriate power allocation scheme can lead to a 1.5x performance gain and 100% EPU when compared with the uniform power allocation strategy routinely used by the homogeneous datacenters.
- We propose GreenHetero, a dynamic power allocation framework that enables adaptive power allocation among heterogeneous datacenter servers to achieve the best performance when the renewable power varies. Specifically, GreenHetero builds a performance-power database for each server configuration and workload type through lightweight online profiling method. Also, the database will be updated at runtime. Based on the database and power prediction, a well-designed Solver can find the optimal power allocation ratio.
- We develop an experimental prototype consisting of several types of servers in racks, a simulated solar power generator, and a rack-level battery provision to evaluate our GreenHetero approach. Based experiments driven by tens of representative datacenter workloads, the evaluation shows that our solution can improve the average performance by 1.2x to 2.2x. The performance gain can reach as much as 4.6x for some server configurations. Also, GreenHetero can achieve up to 2.7x more EPU than the baseline policy.

II. Background

In this section, we first present the typical architecture of modern green datacenters. Then we describe the heterogeneity in green datacenters.

A. *Green datacenters*

To alleviate the carbon emission, datacenters are usually provisioned with clean renewable energy. Figure 2 presents the architectural overview of a typical green datacenters. In this design, the server racks are partially powered by renewable power. Specifically, the on-site renewable power supplies such as photovoltaic (PV) and wind are connected to the power distribution unit (PDU) level to provide a dual-power supply of the grid and renewable power rather than integrating the renewable energy into the utility power. This can help decrease the impacts of voltage transients, frequency distortions and harmonics. Compared with the centralized power integration, the distributed integration prevents PDUs from becoming a power delivery bottleneck.

However, the intermittent and time-varying nature of renewable energy is one of the greatest challenges facing its integration into datacenters. For example, photovoltaic (PV) solar energy is only available during the day and the amount produced depends on the weather and season. When the green power supply is sufficient, it can fully satisfy the power requirement and sustain the server power demand independently. When the green power generation is insufficient, batteries are deployed to supplement the additional power demand as shown in Figure 2. Different from the conventional centralized Uninterrupted Power Supply (UPS) design, for example, Google and Facebook have employed distributed energy storage system [31]. Such decentralized design can avoid single point of failure and increase overall datacenter power availability. Moreover, the centralized UPS battery system lies on the critical power path between the Automatic Transfer Switch (ATS) and the PDU. When UPS is required to manage the power allocation, it supports power transfer for the entire data center but it cannot deal with the power allocation in a fine-grained way. Therefore, the distributed battery can help regulate the power allocation, thus improving the renewable power utilization.

B. *Heterogeneity in green datacenters*

Heterogeneity is prevalent in modern datacenters because servers are gradually provisioned and replaced over the typical 15-year lifetime of a datacenter due to failures, capacity increases, and migrations [28], [7]. Also, to meet

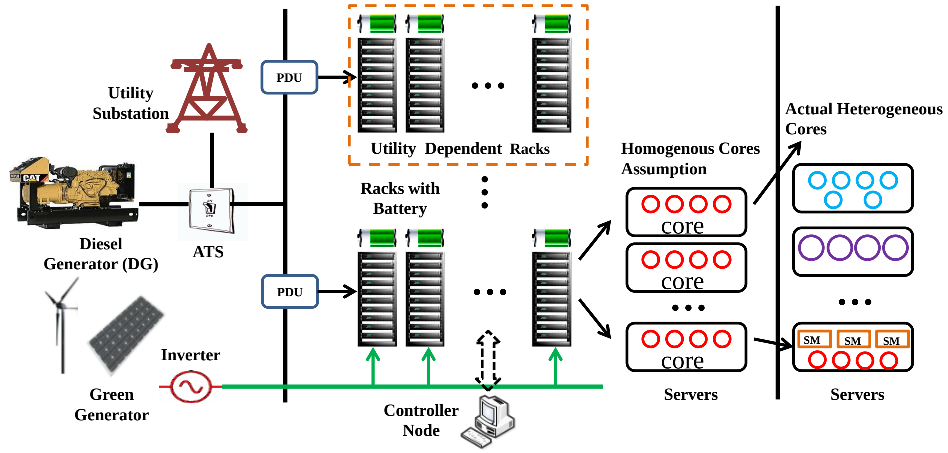


Fig. 2: The architectural overview of a typical green datacenter

the requirement of data processing and computation in the era of big data, users and Internet service providers usually adopt high-performance computing (HPC) platforms, such as FPGAs and GPUs [30]. Over time, this leads to datacenters comprised of a range of heterogeneous platforms with different technologies, power, performance and thermal characteristics, and power management capabilities as shown in Figure 2. When allocating power to these heterogeneous platforms, the power utilization and performance can vary significantly based on the particular allocation that is unaware of datacenter server heterogeneity. However, the diversity across the underlying microarchitectures in datacenters is not explicitly considered by the current datacenter power management system.

III. Heterogeneity analysis in green datacenters

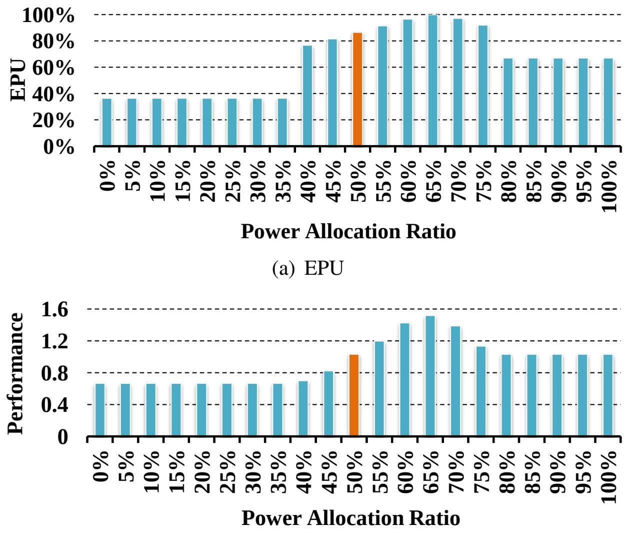
The potential performance benefit of heterogeneity-awareness in power allocation depends on diverse microarchitectural configurations and variable power supply in green datacenters. In this section, we will analyze the impact of server heterogeneity on green power allocation.

A. Effective Power Utilization

To evaluate the effectiveness of a power allocation strategy, we introduce a new metric to measure the effective power utilization, EPU, which can be defined as:

$$EPU = \frac{\sum P_{throughput}}{\sum P_{supply}} \quad (1)$$

where $P_{throughput}$ represents the green power including renewable power and battery energy directly used to generate workload throughput and P_{supply} indicates the current power supply. The value of EPU, between 0 and 1, measures to what extend a power allocation strategy is able to run the servers to their full capacities with the current power supply. Clearly, the closer the EPU value of a power allocation strategy is to 1, the more effective this strategy is to fully utilize the power supply. Different from the Power Usage Effectiveness (PUE) metric that measures the ratio of total amount of energy used by computing equipment, EPU describes how much power is used to generate the computing throughput. EPU can be more accurate to measure the power usage by the servers.



(a) EPU

(b) Performance

Fig. 3: The impact of server heterogeneity on effective power utilization (EPU) and performance. The power allocation ratio (X%) means that X% ((100-X)%) of total current renewable power supply is allocated to Server A (Server B).

B. A case study

We now motivate the need for adaptive power allocation with a case study. We adopt a testbed consisting of two heterogeneous servers to evaluate the performance and EPU. One server has two Intel Xeon E5-2620 processors (denoted as Server A) and the other has an Intel Core-i5 processor (denoted as Server B). The maximum powers measured for the two servers are 81Watt and 147Watt respectively when running the SPECjbb workload. We set a fixed power supply budget of 220Watt, including renewable power and battery energy. In this case, the power supply is insufficient to support the maximal total server power demand.

Figure 3 shows the performance (i.e., throughput *jops* for SPECjbb) and EPU results. The total current renewable power supply is split between Servers A and B and the x-axis indicates the percentage of the total (PAR) allocated to Server A. To highlight the impact of power allocation on performance, the results are normalized to the case of 50% of PAR (i.e., orange bars in these figures). In this case, the power system deploys a uniform power allocation strategy with the assumption of *homogeneous* servers in datacenters. In these two figures, we find that both EPU and performance can achieve best results when the PAR is 65%. As shown in Figure 3(a), the EPU is about 86% using a fair power allocation strategy, which leaves a great potential for improving power utilization. When the PAR equals to 100%, all available power is directed to Server A and the EPU is only about 37%. Figure 3(b) suggests that the performance gain under an appropriate value of PAR is up to 1.5x more effective in power utilization that the uniform strategy.

C. Challenges

The case above suggests several challenges facing the design of an effectiveness of renewable power allocation for heterogeneous green datacenters, which we summarize below.

Platform configurations: Although we only present two types of servers in the above example, there are generally more than two types of server configurations in datacenters as indicated in Figure 1. Needless to say, different server configurations will have different power consumption implications, leading to different optimal PAR values to achieve the best performance.

Workload features: The workload intensity and type will have great impacts on server power demand. For example, the aggregate CPU utilization for a production cluster at Twitter is consistently below 20% [8]. As a result, the power consumption remains at a low level. At the same time, the servers hosting batch workloads such as Mapreduce can consume more power due to the full use of processors. Therefore, we need to recognize different influences on power demand of different workloads.

Intermittent and variable renewable power: The power supply in the above case is fixed and stable. However, the time-varying nature of renewable power supply indeed poses one of the greatest challenges to achieving balanced power allocations. The batteries can be a great supplement when the renewable power is insufficient. However, the unbalanced power discharging activities can also lead to a negative effect on batteries lifetime. We should take both renewable power generation and battery capacity into consideration to improve the effectiveness of power allocation.

Obviously, the optimal power allocation ratio (PAR) actually depends on all of the above factors. To this end, we propose to dynamically identify the optimal operation point at runtime to distribute appropriate power for each rack server.

IV. Design of GreenHetero

In this section, we will introduce the design of GreenHetero, a framework that targets at improving application performance and green power utilization through balanced power allocation for heterogeneous green datacenters. First, we will provide an overview of the GreenHetero controller. Then, we present the details of the adaptive scheduler, which is the key component of GreenHetero.

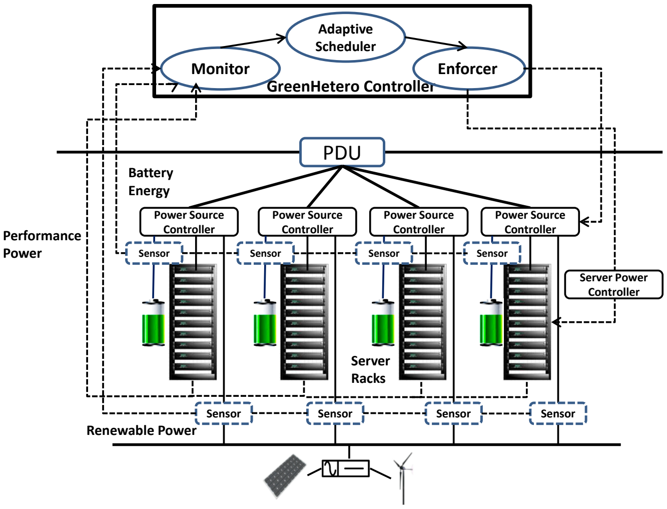


Fig. 4: An overview of the GreenHetero Controller

A. Overview of GreenHetero

Figure 4 shows an overview of the GreenHetero Controller architecture. The GreenHetero controller is a key decision-maker that determines the power source selection and the server power allocation. The controller consists of three main functional modules, including Monitor, Adaptive Scheduler, and Enforcer.

As shown in Figure 4, the Monitor module collects the data on the renewable power generation and the battery energy capacity. The measurements of discharge current of the battery and the available power of renewable power system are gathered by the distributed sensors. Also, it monitors the server-level data, such as application performance and power consumption and reports the data to the scheduler. With the performance and power state feedback, the Scheduler determines the appropriate power source of power supply and the optimal power allocation ratio for the heterogeneous servers (detailed in Section IV-B). With the decision from the Scheduler, the Enforcer is responsible for implementation. The Enforcer has two components: Power Source Controller (PSC) and Server Power Controller (SPC). PSC carries out the switching between power sources. Specifically, servers can be powered by three kind of power sources: renewable power, utility grid power, and battery. SPC controls server power demand using server power state tuning techniques, such as Dynamic Voltage and Frequency Scaling (DVFS).

The reasons for placing the GreenHetero Controller at the rack level in a distributed deployment are threefold. First, it matches the design of the rack-level energy storage system and can explore the full potential of such design. However, the disadvantage of this approach is that the renewable power and energy storage systems for each rack of heterogeneous servers are independent and cannot share their capacities. Second, the distributed rack-level power controller can provide a fine-grained way to track load variability and perform a precise power allocation to guarantee the performance target because the rack-level provisioning will face more load variations than the cluster-level strategy. Third, in the case of power imbalance, the distributed controllers can allow the hotter nodes to be given a proportionally higher power allocation levels to improve the performance while the vast majority of the nodes will run at a lower power allocation level. It has a much higher potential for improving effective power utilization than the uniform cluster-level deployment.

B. GreenHetero Scheduler

In this section, we will present the design details of GreenHetero Scheduler, which attempts to address several research questions. First, how to become aware of datacenter heterogeneity, in terms of both server configurations and workload types? Second, how to determine the power allocation schemes for heterogeneous servers? Third, how to optimize the scheduler? As shown in Figure 5, a power predictor is designed to predict the renewable power supply and the power demand. The objective of this component is to help the scheduler determine the appropriate power sources. Also, based on the profiling data on power and workload, we create and maintain a database, which contains the relationship between server power demand and its performance. Then, the problem solver of this scheduler determines the power source selections based on the prediction and the near-optimal power allocation schemes based on the database. While the output of the Solver is the ratio value of power allocation, the final decision transferred to each server node is the power tuning instructions, such as the exact frequency level using DVFS. The Decision Output component is responsible for the transformation from power value to frequency level.

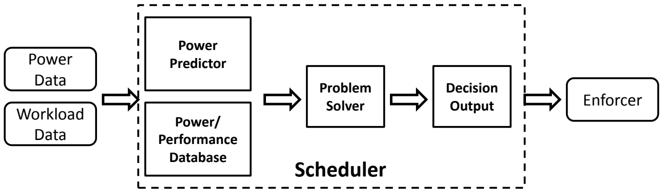


Fig. 5: An overview of the Scheduler

1) Prediction: To decide the power sources in each scheduling epoch (e.g., 15 minutes), we employ a time series prediction technique. Specifically, we leverage the double exponential smoothing prediction (Holt exponential prediction [37]) algorithm to periodically predict renewable power and server rack power, which helps extract meaningful statistical trends and predict future values. This algorithm involves two smoothing equations (Equation 2 for the level and Equation 3 for the trend) and a prediction equation (Equation 4):

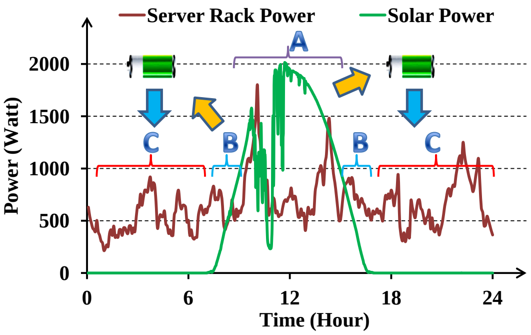


Fig. 6: An illustration of power source selection

$$\text{Level Equation:} \quad S_t = \alpha O_t + (1-\alpha)(S_{t-1} + B_{t-1}) \tag{2}$$

$$\text{Trend Equation:} \quad B_t = \beta(S_t - S_{t-1}) + (1-\beta)B_{t-1} \tag{3}$$

$$\text{Prediction Equation:} \quad P_{t+1} = S_t + B_t \tag{4}$$

where S_t denotes an estimation of the level of the series at time *t*, B_t denotes an estimation of the trend of the series at time *t*, α is the smoothing parameter for the level, $0 \leq \alpha \leq 1$, and β is the smoothing parameter for the trend, $0 \leq \beta \leq 1$. O_t is the observed data from the Monitor. P_{t+1} represents the predicted value for the time epoch *t+1*. We obtain α and β by training the past renewable power generation records. The optimization objective of this training process is to minimize the square difference ΔD^2 between the predicted and observed values as:

$$\mathit{Minimize}: \quad \Delta D^2 = f(\alpha, \beta) \tag{5}$$

in the range constraint of α and β. Note that we select a time series prediction method that is effective for the data center power consumption patterns, but any other proven prediction approaches can be integrated into our prediction framework.

Based on the prediction results, the power source selection process must deal with three unique cases, A, B and C, as shown in Figure 6 that illustrates a typical datacenter server rack power pattern [13] and a 24-hour solar power trace. In Case A, the renewable power is sufficient to independently sustain the power demand of the server rack. Then the surplus power can be used to charge the battery. However, the renewable power is by its own nature fluctuating and depends on the weather conditions, a situation represented by Case B. In this case, the green power supply temporarily drops below the demand and needs the supplement from the batteries, leading to a joint power supply from renewable power and battery energy. Once the battery kicks in, an appropriate power allocation strategy becomes more important because the unbalanced power discharging activities can reduce the battery lifetime and decrease the energy efficiency. If the renewable power becomes completely unavailable, a scenario illustrated by Case C, the battery independently sustains the power demand. In all conditions, the grid power will be the last resort only when the battery drains out. Note that, when the grid is used to power all server racks, including utility-dependent and renewable-power-dependent racks, the grid power budget allocated to each server rack will be reduced. For the sake of battery lifetime and energy efficiency, we make the following assumptions. First, we set the depth of discharging (DoD) at 40% to mitigate the impact on battery lifetime. Second, when the batteries reach the preset DoD, the grid or the renewable power will charge the batteries to prepare for future power shortages. Third, there is only one power source that can charge the battery at any given time.

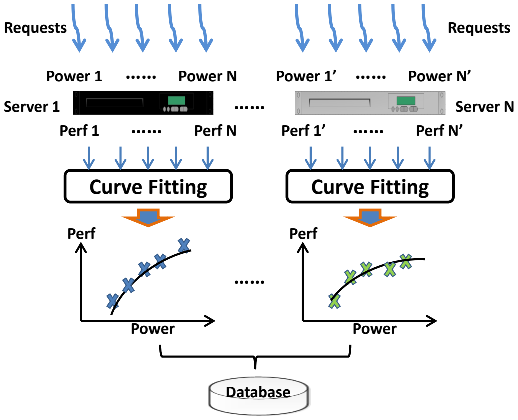


Fig. 7: Populating and updating the database

2) Database: The adaptive power allocation is a technique to automatically find the near-optimal allocation ratio from time-varying renewable power supply to the processing servers for the given application and server configuration. First, in order to effectively and timely adapt to the underlying server heterogeneity and workload variability, we present a performance-power database through lightweight on-line profiling.

The Database is designed as a guideline for the Solver and it provides the power consumption and throughput projection for all workloads and server configurations it has ever executed. Instead of the exhaustive method that introduces a huge amount of off-line profiling, we build the database using a *training run* process, as depicted in Figure 7. The first time that a workload arrives at the server system, it will be executed with enough power supply to minimize the performance overhead. Specifically, we use the system power governor as *ondemand*, which can dynamically adjust the frequency level of processor according to instantaneous CPU utilization, to guarantee the workload performance. The battery and grid power will always be ready to support the power demand during the training run in case of the power shortages due to the renewable power fluctuations. The duration of training run, typically 10 minutes, is slightly shorter than the scheduling epoch of 15 minutes. In the training run, the performance (Perf 1, ..., Perf X) and power usage (Power 1, ..., Power X) data on each type of server will be collected and then written into the database every 2 minutes. With the limited profiled data, GreenHetero then uses curve fitting to construct relational equations *Perf* $= f($*Power*$)$ as a performance projection for each workload on each server platform.

3) Problem Solver: To achieve the optimization objective, we first formulate the optimization problem. For simplicity, we continue to use two types of servers as an example, Server A and Server B. Then the performance of each server can be denoted as:

$$\text{Server A:} \quad Perf_t^a = f(l_a, m_a, n_a, Power_t^a) \tag{6}$$

$$\text{Server B:} \quad Perf_t^b = f(l_b, m_b, n_b, Power_t^b) \tag{7}$$

where $Power_t^N$ is the allocated power for Server N at time t within the range of server peak power and idle power. When the $Power_t^N$ is greater than the peak power, the performance will stay constant. When it is less than the idle power, the performance value will become zero. l_N, m_N, and n_N are constant real numbers for Server N derived from the curve fitting process. Specifically, we adopt nonlinear curve fitting (i.e., quadratic curve) here to extract the relationship between power provision and performance for two reasons. First, when the power supply exceeds the maximum server power value, the performance will not increase any more. Therefore, the linear curve projection is not suitable in such cases. Second, we use a quadratic relational function within the power demand range to reduce the complexity of the solver compared with higher order functions while minimizing the error compared with linear function.

The objective of the Solver is to maximize the overall performance of rack servers by finding the optimal value of power allocation ratio (PAR). The process of solving the problem is similar to prior work [24]. We denote η as the PAR of Server A and γ as the PAR of Server B, where $0 \leq \eta + \gamma \leq 1$. As a result, we arrive at Equation 8:

$$\begin{aligned} Perf_t &= maximize\ (Perf_t^a + Perf_t^b) \\ &= maximize\ (f(l_a, m_a, n_a, \eta Power_t) + \\ & f(l_b, m_b, n_b, \gamma Power_t)) \end{aligned} \tag{8}$$

where $Power_t$ is the predicted power supply for time t derived from Equation 4. Note that, when there are more than one Server A and Server B, we distribute the same amount of power to the same type of servers by default. For example, if the rack consists of x Server As and y Server Bs, then the PAR of each Server A is $\frac{\eta}{x}$ and Server B is $\frac{\gamma}{y}$. Further, when there are more than two types of servers such as three types of servers as we can find in Figure 1, in which 80% of datacenters consist of two and three types of server configurations, we will add another PAR variable δ in Equation 8. Note that, the extra ratio $(1-\eta-\gamma)$ of power supply will be charged into batteries if the renewable power generation is sufficient. In this work, we focus on the power allocation management at the rack level, so we assume the server configurations in each PDU racks can be up to three types and put more complex cases as future work.

4) Output Decision: Once the PAR is determined, the scheduler is able to allocate the power to each server. At the server level, the power consumption can be controlled by the power state such as DVFS and power saving state (e.g., Sleep and Hibernation) [14], [33]. Obviously, when the power supply exceeds the peak server power demand, we directly set the frequency of all processor cores at the highest level. We create a power state set for each kind of server S_N, which consists of all server frequency levels and low power states and is ordered from low power state to high power state. We use a mapping scheme that relates the power output and the server power state. We set the minimum and maximum values of the power range, and any value between the power limits is linearly scaled to a position in the state set S_N.

Algorithm 1 Optimization Algorithm

```
1: // At the beginning of each scheduling epoch t
2: Obtain current server configuration c and workload type w, then search
   the database.
3: If ( c & w == 0) Then     //Does not find a relational equation for
   Server c running workload w in database
4:     Step into Training run process
5:     Add new relational projection into database
6: Else     //Find the relational equation and update the existing database
7:     Find the optimal PAR by the Solver
8:     Continue the execution and collect the result data
9:     Reconstruct the relational equation using curve fitting with both new
   and old profiling data
10:    Update the database
11: EndIf
```

5) Optimization on Power Allocation: Although we can get a relational equation using the database as mentioned above, the scheduler cannot always guarantee the optimal power allocation results because the information from the profiling data is limited in the training run and can be less accurate. Therefore, the database needs to be dynamically updated. Algorithm 1 shows the pseudo code of two key optimization operations, i.e., adding new relational projections (line 4-5) and updating existing database (line 7-10). At the beginning of each scheduling epoch, the scheduler first obtains current server configurations and workload types. If the relational equation does not exist in the database when searching the database, it will step into the *training run* process and add the new projection into the database. If the performance and power projections based on the current server configuration and workload type can be found, then use the Solver to find the optimal PAR. The feedback performance and power consumption results will be used to reconstruct the relational equation along with the old profiling data. Finally, the new equation will be written into the database and the updating process ends.

V. Evaluation of GreenHetero

A. Methodology

1) Evaluation Workloads: To conduct a meaningful and fair evaluation of GreenHetero, we choose various datacenter workloads from Cloudsuite [25], PARSEC [6], SPEC [3], [2], and Rodinia [29], summarized in Table I. Cloudsuite is a benchmark suite for typical cloud services, such as Web-search and Memcached. PARSEC focuses on emerging workloads, such as computer vision, video encoding, financial analytics, animation physics and image processing. SPECCPU represents high performance computing (HPC) applications and Rodinia is designed for GPU-CPU heterogeneous computing. Within each experiment, a workload can be executed iteratively.

Workloads	Suite	Performance Metric
SPECjbb	SPEC [3]	jops (99%-ile 500ms constrained)
Web-search, Memcached	Cloudsuite [25]	ops (90%-ile 500ms constrained) rps (95%-ile 10ms constrained)
Streamcluster, Freqmine, Blackscholes, Bodytrack, Swaptions, Vips, X264, Canneal	PARSEC [6]	ips (instructions per second), execution time
Mcf	SPECCPU [2]	ips, execution time
Srad_v1, Particlefilter, Cfd, Streamcluster	Rodinia [29]	ips, execution time

TABLE I: Workload Description

Server Type	Frequency	Socket	Cores	Peak Power	Idle Power
Xeon E5-2620	2.0 GHz	2	12	178W	88W
Xeon E5-2650	2.0 GHz	1	8	112W	66W
Xeon E5-2603	1.8 GHz	1	4	79W	58W
Core i7-8700K	3.7 GHz	1	6	88W	39W
Core i5-4460	3.2 GHz	1	4	96W	47W
Nvidia Titan Xp	1582 MHz	1	3840	411W	149W

TABLE II: Server Description

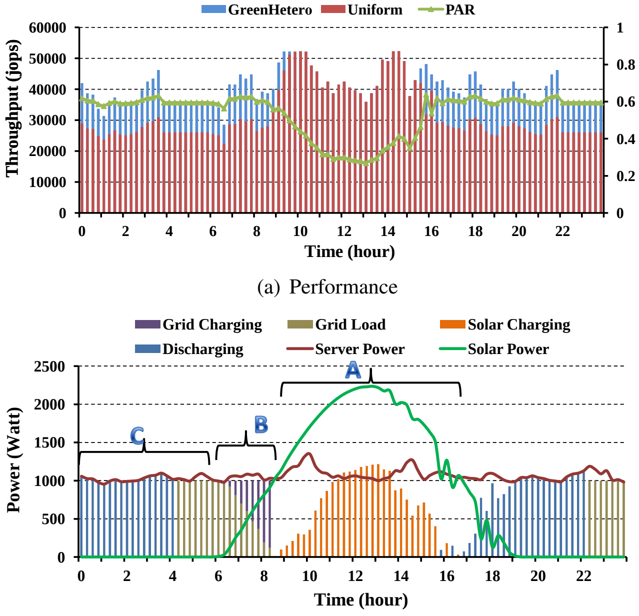
2) Evaluation Platform : The different workloads can run on different server platforms. In our evaluation, we consider 6 different configurations involving Intel CPUs and a Nvidia GPU and each configuration consists of 5 servers in racks. The specific configurations are presented in Table II. Each server runs on the Ubuntu Linux OS with the kernel version 3.0.13. To ensure the consistency, each server has 32 GB main memory. The server racks share the storage through a network file system. The power consumption of each server is monitored by an external power meter [1]. We use the *cpufreq and nvidia-smi* command to scale CPU and GPU frequencies respectively. Also, *perf and nvprof* command can be used to obtain the performance data.

To simulate a data center with renewable energy provision, we choose two of the renewable power production traces with one-week duration from NREL [5], including irradiation every 15 minutes, and replay the chosen trace on our prototype. Specifically, one of the solar traces, referred as *High trace*, represents the high level renewable power generation and the other one, *Low trace*, represents the low level renewable power generation. We use 10 12V 100Ah lead-acid batteries for the server racks, which can store the surplus renewable energy. We also assume a DoD (depth of discharging) of 40% in our setup, which translates to a lifetime of 1300 recharge cycles [31]. The choice for energy capacity of the battery considers both the renewable energy production and energy consumption of the server racks. The energy efficiency of the battery is set at 80% as detailed in [12].

3) Power Allocation Policies and Metrics : To be more specific, we compare *GreenHetero* to four different power management policies summarized in Table III. *Uniform* is a heterogeneity-oblivious policy that always allocates power to each server uniformly. *Manual* statically tries all possible

Policies	Description
Uniform	Allocate power to each server uniformly without considering server heterogeneity and workload type
Manual	Determine the near-optimal ratio by trying all possible power allocations at a granularity of 10%
GreenHetero-p	Allocate power to the server based on the order of energy efficiency
GreenHetero-a	Determine the power allocation ratio as GreenHetero without optimizations
GreenHetero	Determine the power allocation ratio adaptively at runtime

TABLE III: Description of Power Allocation Policies



(a) Performance

(b) Power Profile

Fig. 8: Runtime results of SPECjbb using *High solar trace*

power allocations at a granularity of 10%. *GreenHetero-p* determines the power allocation based on the descending order of the energy efficiency (i.e., throughput per watt), which is derived from the Database of *GreenHetero*. *GreenHetero-a* is a simplified version of *GreenHetero* without the optimization of dynamically updating database. *Uniform* is regarded as the baseline policy. The main metrics are workload performance, i.e., runtime throughput such as jops for SPECjbb, and EPU.

B. Results and Discussions

In this section, we present the detailed performance and utilization comparisons of GreenHetero with other four baseline power management policies under different workload conditions.

1) Runtime Results: We first show the results of a 24-hour run of SPECjbb using typical datacenter workload pattern (shown in Figure 6). Also, the *High solar trace* is used for the evaluation. We initialize the battery capacity to its maximal state at the starting time. When the grid power is used to supply the power, the power budget is set at 1000W, which is lower than the server power demand as aforementioned. In the following part, we focus on two fixed server configurations, i.e., 10 total servers equipped with E5-2620 and Core-i5 processors. The power allocation ratio (PAR) indicates the percentage of power allocated to server with E5-2620. Figure 8 presents the performance and power profile results of the GreenHetero and Uniform policies. As shown in Figure 8(a), GreenHetero outperforms the Uniform policy for most scheduling epochs. On an average, GreenHetero achieves up to 1.5x performance gain when the renewable power supply is insufficient (i.e., *Case B* and *C* denoted in Section IV-B). When the renewable power supply is sufficient, the performance of GreenHetero is similar to Uniform, suggesting that adaptive power allocation has very little impact when the power supply is abundant. With the power supply varying, the PAR changes accordingly. The average value of PAR during the 24-hour execution is about 58%. To achieve the best performance, the scheduler has to dynamically adjust the ratio value based on the power supply and server power demand.

Figure 8(b) shows the discharging and charging activities of the batteries. In *Case C*, the batteries continuously discharge to supplement the unavailable renewable power until their DoD is reached, which lasted for about 4.2 hours. When the batteries can no longer sustain the power demand, the grid will take over and support the server power demand (*Grid Load*), and charge the battery (*Grid Charging*). In *Case A*, the renewable power can independently support the load and the extra power will be charged into the batteries.

2) Impact of Workload types: To understand the impact of different workloads, the evaluation results, in terms of performance and effective power utilization (EPU), of 13 different workloads under five power allocation policies are reported here. Specifically, to explore the importance of power allocations, we focus on the analysis of the case when the renewable power is insufficient.

Performance: Figure 9 shows the performance results of 3 interactive datacenter workloads, 8 PARSEC workloads, and 1 HPC workload. Overall, GreenHetero performs the best among the five policies, achieving an average of 1.6x performance gain over the baseline Uniform policy. The *Streamcluster* and *Memcached* workloads show the best and worst performance gains by up to 2.2x and 1.2x respectively. Despite of the fact that the Manual policy allocates the power by a 10% granularity and its PAR accuracy is very low, it still performs better than the Uniform policy. GreenHetero-p will always distribute power to the server with Core-i5 first due to its higher energy efficiency. This policy works when compared with Uniform and some cases of Manual. However, it depends on the specific power supply and server configurations. For example, if the rest of the power supply cannot support the other server to power on, then the power allocation will be unbalanced, further wasting as evidenced by the *Streamcluster* workload. GreenHetero-a uses limited power and performance profile to construct the database without updating. In some cases, GreenHetero-a underperforms GreenHetero that has dynamically optimizations. Therefore, the optimizations are also important and can provide additional

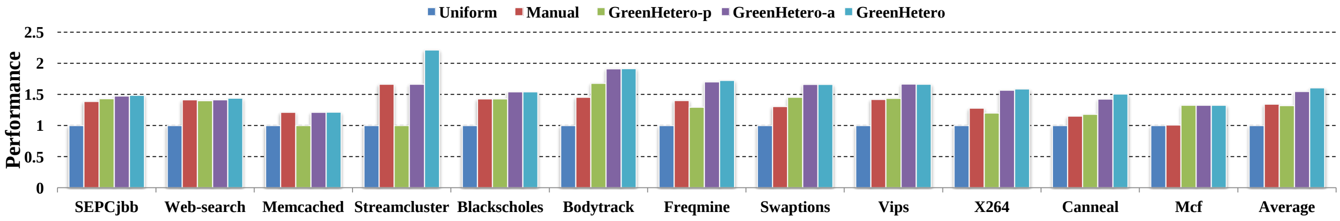


Fig. 9: Performance of five power allocation policies for different workloads

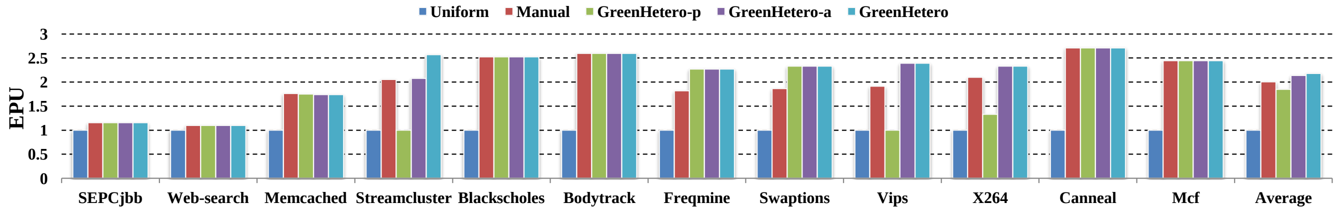
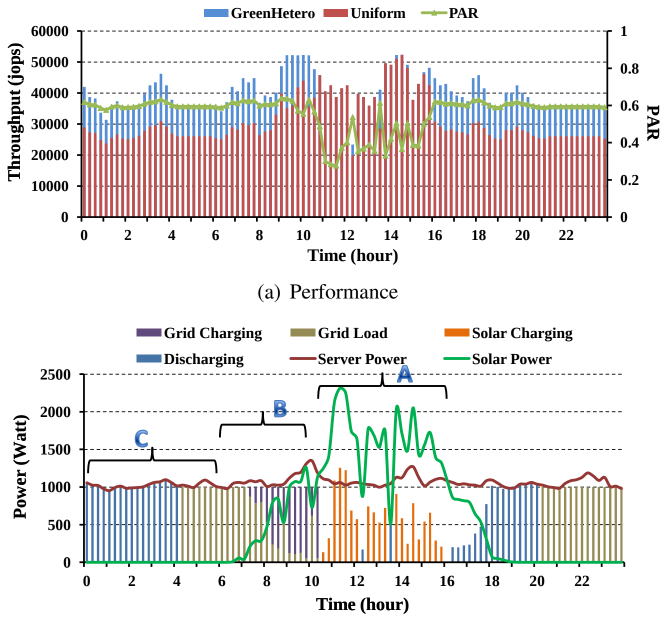


Fig. 10: Effective power utilization of five power allocation policies for different workloads

performance improvement. The HPC workload Mcf shows a 1.3x performance gain by GreenHetero, which is similar to GreenHetero-a and GreenHetero-p.

EPU: Figure 10 shows the EPU results. The average EPU achieved by GreenHetero is about 2.2x higher than Uniform. In many cases, the other four policies introduce the same EPU (normalized to Uniform). The *Canneal* workload shows the best result by up to 2.7x improvement while the *Web-search* workload demonstrates only 1.1x improvement. We find that the EPU value does not have any specific correlation with the performance value, other than an evident trend indicating that an increasing EPU tends to correlate to an improved overall performance and reduced inefficiency power allocations. Therefore, EPU is a very important metric to evaluate the effectiveness of the power allocation.

3) Impact of renewable power supply: In what follows, we evaluate another renewable power generation pattern for SPECjbb. Different from the *High solar trace* in Figure 8, we use the *Low solar trace* here to analyze the impact of different power generation patterns. Compared with the former, the power supply in the latter becomes more fluctuated. As shown in Figure 11(a), the performance of Uniform is consistently lower than that of GreenHetero when the renewable power supply is not in abundance. On an average, GreenHetero can still achieve 1.2x performance gains over Uniform during those epochs in Cases A and B. With its adaptive adjustment of the PAR value, GreenHetero is able to fully explore the potentials of the time-varying power supply. Figure 11(b) presents the power profiles of GreenHetero. Obviously, compared with *High solar trace*, *Low solar trace* shows more frequent discharging/charging activities. In the evaluation, GreenHetero discharge the batteries twice per day (to the maximum DoD), so there is relatively very small impact on the lifetime. However, the rest of renewable power after powering the server racks may not fully charge the battery, leading to more grid power cost (e.g., more grid power usage over a 4-hour period than when *High solar trace* is used).



(a) Performance

(b) Power Profile

Fig. 11: Runtime results of SPECjbb using *Low solar trace*

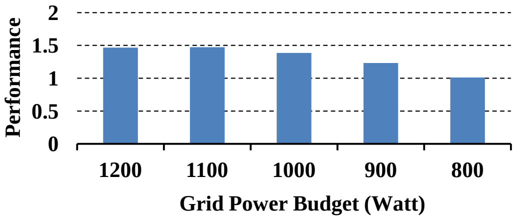


Fig. 12: Performance of different grid power budget

4) Impact of grid power budget: Figure 12 presents the effectiveness of GreenHetero under different grid power

Combinations	Server Types	Workloads
Comb1	E5-2620, i5-4460	SPECjbb
Comb2	E5-2603, i5-4460	
Comb3	E5-2650, E5-2620	
Comb4	i7-8700K, i5-4460	
Comb5	E5-2620, E5-2603, i5-4460	
Comb6	E5-2620, Titan Xp	Streamcluster, Srad_v1, Particlefilter, Cfd

TABLE IV: Server Combinations

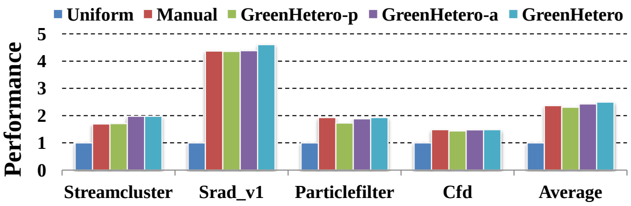


Fig. 14: Performance of *Comb6* for different workloads

budgets when the batteries drain out. As the figure shows, the performance gain for SPECjbb by GreenHetero becomes much lower than Uniform when the power budget decreases. However, due to the high utility charges for peak grid power (e.g., up to $13.61/kW as mentioned in [21]), simply increasing the grid power budget for higher performance can introduce enormous cost. Especially for heterogeneous datacenters, GreenHetero can achieve better effectiveness for power utilization. As a result, GreenHetero is able to further help underprovision the grid power infrastructure, which is orthogonal to prior works [14], [33].

5) Impact of server heterogeneity: In this section, we evaluate the impact of different combination of server configurations (based on Table II), summarized in Table IV. We show the results of several representative workloads.

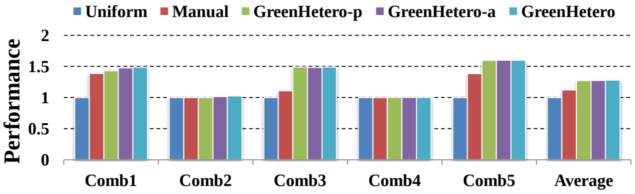


Fig. 13: Performance of different server combinations

Figure 13 presents the results of SPECjbb for different server combinations. As we expect, the individual server configuration has a great impact on the performance. Comb2 and Comb4 show similar results (only 3% improvement) for all policies because these two pairs of servers have similar power profiles. That is, these server configurations exhibit behaviors consistent with those of *homogeneous servers* when running the SPECjbb workload. In this case, the advantage of GreenHetero is diminished. Meanwhile, Comb1 and Comb3 present up to 1.5x performance gains over Uniform. These servers exhibit truly *heterogeneous* behaviors and show large difference in energy efficiency. As a result, we believe that GreenHetero can provide even greater benefits for datacenters with higher level of heterogeneity in server configurations. We also evaluate the combination for three server configuration, i.e., Comb5. GreenHetero achieves 1.6x performance gains over Uniform. This result also depends on the specific server configurations.

We also conduct several experiments on a GPU-based platform. The results are shown in Figure 14. Generally, GreenHetero performs the best among all policies. Due to the large difference in computing performance between GPU and CPU, the performance improvement by GreenHetero can be up to 4.6x for the *Srad_v1* workload. The average performance gain is 2.5x. For *Cfd*, the performances running on CPU and GPU are similar. Therefore, the overall performance gain is not as high as *Srad_v1*. We can conclude that it is extremely essential to design such heterogeneity-aware power allocation policy for GPU-based server configurations in green datacenters.

VI. Related work

Power Management in Datacenters: There have been many recent studies on the power management in datacenters [17], [9], [11], [10], [20], [19], [26], [21], [18], [27]. However, they all assume that the computing environments are homogeneous and their proposals are not aware of server heterogeneity. For example, green datacenter design of *Oasis* uses Intel Core i7-2720QM 4-core CPU as processing engine [11]. GreenSlot and GreenHadoop uses a 16-server cluster, where each server is a 4-core Xeon [20], [19]. GreenPar uses 55 servers, where each server is equipped with a dual-core 1.6GHz Atom processor [26]. Different from these works, GreenHetero conducts power management with an awareness of the underlying heterogeneous servers. GreenGear incorporates wimpy servers into existing green datacenters to dynamically deal with power mismatches [34]. GreenHetero differs from GrearGear in the following important aspects. First, the GreenGear design introduces extra capital cost by purchasing new and better energy-efficient servers, while GreenHetero tries to mitigate the side effect of datacenter upgrading that leads to the increased server heterogeneity. Second, GreenGear targets at solving the power mismatch problem, while GreenHetero aims to improve the server performance and renewable power utilization by achieving effective and efficient power allocations. Third, GreenGear adopts an *on-off* server strategy and always turns on only one server in each composite heterogeneous node. Obviously, when the power supply is sufficient, *all-on* strategy can be more effective. As a result, GreenHetero is suitable for all cases and can adaptively adjust the power allocation policy under different server configurations and workloads.

Heterogeneity in Datacenters: There have also been studies that explore the heterogeneity in datacenters [22], [7], [15]. Whare-Map exposes and quantify the performance impact of the heterogeneous datacenters and performs job-to-machine mapping [22]. Paragon is an online and scalable datacenter scheduler that is heterogeneity and interferenceaware to improve server utilization [7]. KnightShift is a server-level heterogeneous architecture that introduces an active low power mode, along with a high-power primary server, to improve energy proportionality [15]. Different from these works that emphasize on the impact of heterogenous servers on server utilization and energy efficiency, GreenHetero focuses on the impact of datacenter heterogeneity on renewable power allocations and performance.

VII. Conclusion

In this paper, we propose GreenHetero, a dynamic power allocation framework that enables adaptive power allocations to achieve the best performance when the renewable power supply varies in its availability. GreenHetero maintains a database for each server configuration and workload type. A well-designed Solver can find the optimal power allocation ratio. Using representative datacenter workloads, the evaluation shows that our solution can improve the average performance by 1.2x to 2.2x.

Acknowledgment

We would like to thank the anonymous reviewers for their valuable feedback and suggestions. This work was supported in part by National key research and development program of China under Grant 2018YFA0701800, NSFC No. 61821003, and NSFC No. 61872156, the US NSF under Grant No.CCF-1704504 and No.CCF-1629625. Haoran Cai has graduated from Huazhong University of Science and Technology and joined Huawei Technologies Co., Ltd. and this work is partly supported by Algorithm and Technology Development Department, Carrier BG. We want to thank new colleagues for their guidance and suggestions.

References

[1] Zh-101 portable electric power fault recorder and analyzer, 2009.

[2] SPECCPU. http://www.spec.org/cpu2006/., 2010.

[3] SPECJBB 2013:Java Business Benchmark. http://www.spec.org/jbb2013/, 2014.

[4] Critical Action Needed to Save Money and Cut Pollution. https://www.nrdc.org/, 2015.

[5] Measurement and instrumentation data center. http://www.nrel.gov/midc/, 2015.

[6] C. Bienia et al. The parsec benchmark suite: Characterization and architectural implications. In *PACT*, 2008.

[7] C. Delimitrou et al. Paragon: Qos-aware scheduling for heterogeneous datacenters. In *ASPLOS*, 2013.

[8] C. Delimitrou et al. Quasar: resource-efficient and qos-aware cluster management. In *ASPLOS*, 2014.

[9] C. Li et al. Solarcore: Solar energy driven multi-core architecture power management. In *HPCA*, 2011.

[10] C. Li et al. iswitch: coordinating and optimizing renewable energy powered server clusters. In *ISCA*, 2012.

[11] C. Li et al. Enabling datacenter servers to scale out economically and sustainably. In *MICRO*, 2013.

[12] C. Li et al. Enabling distributed generation powered sustainable high-performance data center. In *HPCA*, 2013.

[13] D. Wang et al. Energy storage in datacenters: what, where, and how much? In *SIGMETRICS*, 2012.

[14] D. Wang et al. Underprovisioning Backup Power Infrastructure for Datacenters. In *ASPLOS*, 2014.

[15] D. Wong et al. Knightshift: Scaling the energy proportionality wall through server-level heterogeneity. In *Micro*, 2012.

[16] G. Boccaletti et al. How it can cut carbon emissions. *McKinsey Quarterly*, 2008.

[17] H. Cai et al. Greensprint: Effective computational sprinting in green data centers. In *IPDPS*, 2018.

[18] H. Zhang et al. Maximizing performance under a power cap: A comparison of hardware, software, and hybrid techniques. In *ASPLOS*, 2016.

[19] I. Goiri et al. Greenslot: scheduling energy consumption in green datacenters. In *SC*, 2011.

[20] I. Goiri et al. Greenhadoop: leveraging green energy in data-processing frameworks. In *EuroSys*, 2012.

[21] I. Goiri et al. Parasol and greenswitch: Managing datacenters powered by renewable energy. In *ASPLOS*, 2013.

[22] J. Mars et al. Whare-map: heterogeneity in homogeneous warehouse-scale computers. In *ISCA*. ACM, 2013.

[23] L. Barroso et al. The datacenter as a computer: An introduction to the design of warehouse-scale machines. 2013.

[24] L. Liu et al. Heb: deploying and managing hybrid energy buffers for improving datacenter efficiency and economy. In *ISCA*, 2015.

[25] M. Ferdman et al. Clearing the clouds: a study of emerging scale-out workloads on modern hardware. In *ASPLOS*, 2012.

[26] M. Haque et al. Greenpar: Scheduling parallel high performance applications in green datacenters. In *ICS*, 2015.

[27] Q. Wu et al. Dynamo: facebook's data center-wide power management system. In *ISCA*, 2016.

[28] R. Nathuji et al. Exploiting platform heterogeneity for power efficient data centers. In *ICAC*, 2007.

[29] S. Che et al. Rodinia: A benchmark suite for heterogeneous computing. In *IISWC*, 2009.

[30] S. Mittal et al. A survey of methods for analyzing and improving gpu energy efficiency. *CSUR*, 2015.

[31] V. Kontorinis et al. Managing distributed ups energy for effective power capping in data centers. In *ISCA*, 2012.

[32] X. Qu et al. Optimatch: Enabling an optimal match between green power and various workloads for renewable-energy powered storage systems. In *ICPP*, 2017.

[33] X. Zhou et al. Underprovisioning the grid power infrastructure for green datacenters. In *ICS*, 2015.

[34] X. Zhou et al. Greengear: Leveraging and managing server heterogeneity for improving energy efficiency in green data centers. In *ICS*, 2016.

[35] Y. Zhang et al. Greenware: Greening cloud-scale data centers to maximize the use of renewable energy. In *Middleware*, 2011.

[36] Apple Inc. Apple environmental responsibility report. https://www.apple.com/environment/reports/docs, 2014.

[37] P. Kalekar. Time series forecasting using holt-winters exponential smoothing. 2004.

[38] Data Center Knowledge. Data centers scale up their solar power. http://www.datacenterknowledge.com, 2012.