Green-Aware Workload Scheduling in Geographically Distributed Data Centers

Changbing Chen, Bingsheng He, Xueyan Tang
School of Computer Engineering
Nanyang Technological University, Singapore
Email: {chchangb, bshe, asxytang}@ntu.edu.sg

***Abstract*—Renewable (or *green*) energy, such as solar or wind, has at least partially powered data centers to reduce the environmental impact of traditional energy sources (*brown* energy with high carbon footprint). In this paper, we propose a holistic workload scheduling algorithm to minimize the brown energy consumption across multiple geographically distributed data centers with renewable energy sources. While green energy supply for a single data center is intermittent due to daily/seasonal effects, our workload scheduling algorithm is aware of different amounts of green energy supply and dynamically schedules the workload across data centers. The scheduling decision adapts to workload and data center cooling dynamics. Our experiments with real workload traces demonstrate that our scheduling algorithm greatly reduces brown energy consumption by up to 40% in comparison with other scheduling policies.**

***Keywords*-Green data centers, renewable energy, workload scheduling, geographically distributed data centers.**

I. Introduction

Data centers have been the key system infrastructure for cloud computing. Many large IT companies such as Google and Microsoft have multiple geographically distributed data centers. Data centers are significant energy consumers due to not only their computing equipments but also cooling and other facilities. It has been shown that worldwide data centers run the risk of doubling their energy consumption every 5 years [17]. The high energy footprint of data centers leads to serious environmental issues (including e-waste and CO_2 emission). There has been a tremendous amount of efforts in research and development to resolve those environmental issues. Most techniques (e.g., [4], [7]) target at reducing the energy consumption in order to reduce the environmental impact of traditional energy sources (*brown* energy). Recently, with increasing adoption of renewable energy supply techniques (such as solar panels) [9], data centers have been powered at least partially with green energy [29]. This paper aims at minimizing the brown energy usage by utilizing the green energy available in multiple geographically distributed data centers.

Research interests have been growing in integrating renewable energy into data centers (e.g., [11], [10], [18], [34], [24]). The key challenge of such an integration is that green energy sources in a limited area are variable and intermittent due to daily/seasonal effects. On the other hand, workloads to data centers fluctuate significantly [14], [13]. We have observed the significant mismatch between workloads and green energy supply. When the workload is at its peak, green energy supply sometimes can be nearly zero, and vice versa. This mismatch challenges the utilization of green energy and thus potentially increases the brown energy usage.

While existing studies [11], [10], [18], [34], [24] have demonstrated their effectiveness in addressing the mismatch between workload and green energy supply, they are suboptimal in minimizing the brown energy usage in multiple geographically distributed data centers. For the studies [11], [10], [18] that are limited to a single data center, they by design cannot fully address the mismatch problem. Instead, multiple geographically distributed data centers allow more flexibility in utilizing renewable energy. Take solar energy as an example. Data centers belonging to the same company can be located in different time zones. The amount of solar energy generated at different data centers can be complementary to each other. For the studies [34], [24] that run across multiple data centers, they are limited to online service workloads. Online service workloads do not have sufficient slacks in allowing more advanced scheduling to resolve the mismatch. Moreover, existing studies on either single or multiple data centers have ignored the connection between renewable energy and physical environment. Specifically, the outside temperature significantly affects the cooling energy consumption. On the other hand, the outside temperature is usually higher when solar energy supply is higher (e.g., on hot sunny days). The ignorance of this connection leads to wrong decisions of scheduling.

To address the aforementioned issues, we propose a holistic workload scheduling algorithm (called *MinBrown*) to minimize the brown energy consumption across multiple geographically distributed data centers all with renewable energy sources. It targets at the workloads with reasonable slacks (particularly, scientific workloads, high performance computing tasks). Overall, our scheduling algorithm is holistic in taking advantage of a series of key factors in green energy usage: the variable green energy supply, outside temperature and cooling energy consumption in each data center involved, and workload fluctuation, deadline and job structure. Specifically, given the user specified deadline, the workloads are dynamically scheduled to the data center

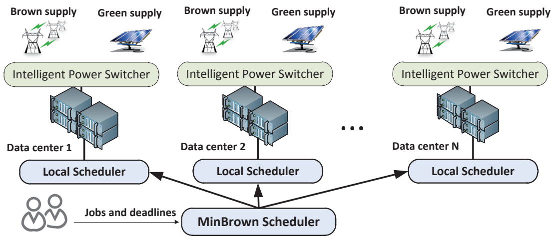


Figure 1. Geographically distributed data centers with both green and brown energy sources.

where the solar energy supply best satisfies the energy demand. Moreover, the scheduling considers the resultant cooling energy consumption when assigning the workload. It also supports fine grained workload placement and migration via virtual machine operations.

We evaluate our workload scheduling algorithm with simulations. The simulator takes real traces on workloads and solar energy generation as input. We compare the scheduling algorithm with green-oblivious approach (round-robin) and other green-aware approaches. With extensive experiments, we demonstrate that our scheduling algorithm achieves up to 40% and 21% less brown energy than green-oblivious and other green-aware approaches, respectively. Moreover, it achieves almost the same amount of total energy consumption as other approaches.

Organization. The reminder of this paper is organized as follows. Section II formally defines the problem. Section III presents our scheduling algorithm, followed by experimental results in Section IV. We review the related work in Section V and conclude this paper in Section VI.

II. Problem Definition

Data centers. This paper considers public cloud providers (like Google and Microsoft) which run their services in geographically distributed data centers, as illustrated in Figure 1. Virtualization is used to manage the computation resource. The basic unit of resource allocation is virtual machine with the predefined amount of CPU/DRAM/Disk resource. Each data center has its nearby green sources either built by the data center owner or other utility company.

Each data center has a switch connected with both green sources and brown sources (e.g., public grids). Research has been devoted to improve the effectiveness of this kind of switch [23]. This paper assumes an ideal switch: when the power demand is higher than green power supply, it immediately draws power from brown sources. Otherwise, the surplus green energy is used for other purposes. We assume that batteries are used for emergency purposes only. Previous studies [10], [11], [18], [2] also have the same assumption.

Each data center has the cooling facilities to avoid overheating in its equipments. The cooling energy consumption depends on not only the data center utilization but also other factors including the cooling strategies and outside temperature. To keep the data center below the predefined temperature, a higher outside temperature generally requires more cooling energy. Moreover, there are a series of discrete levels in cooling strategies (whether a chiller is on or off, a certain level in fan speed, whether air is circulated through chiller). Essentially, the cooling strategies form multiple levels of cooling energy usage, and transitions among different levels result in a leap in the energy consumption [21].

Workload. We study non-interactive workloads which have slacks to exploit more green energy. A workload consists of many jobs, each of which is represented as a directed acyclic graph (DAG). Each node in a DAG represents a task. The job structure allows us to have finer grained scheduling on tasks. Our scheduling algorithm allows users to specify QoS: each job has a specified deadline to define its slack. We focus on scheduling high performance computing (HPC) workloads in this paper, and leave the scheduling of other workloads (like MapReduce [6] and graph processing [5]) as future work.

Persistent data is typically replicated across (a small number of) data centers for availability against disasters. This replication improves availability and also allows flexible workload placement and scheduling. For simplicity, we assume persistent data are replicated in all data centers, and our scheduling algorithm is straightforward to be extended to deal with replication in a subset of data centers only (e.g., [16]). That is, we assume that a job can run on any of the data centers.

Optimization goal. Given a workload submitted to multiple geographically distributed data centers, a workload scheduler assigns and migrates the workload across the data centers. The optimization goal is to minimize the total usage of brown energy in all the data centers, given the constraint that all jobs in the workload are completed within their predefined deadlines.

III. Workload Scheduling Algorithm

This section present the detailed design of our workload scheduling algorithm.

A. *Design Rationale*

Finding the optimal workload scheduling for minimizing the brown energy consumption is a complicated problem. It involves the following categories of parameters. The first category is on green energy supply: the available amount of green energy supply in each data center, which is variable and intermittent along the time. The second category is on data centers: 1) available capacities and the network latency and bandwidth among data centers, which affects the decision on workload scheduling, and 2) equipment and cooling power. The third category is on the workload: the computation requirement of workloads, job structures and their deadline. These parameters are intertwined with each other, making the scheduling decision complicated. For example, when scheduling a task in a target data center, it affects the available amount of green energy of the data center. This decision also affects the green energy usage of other tasks in the same job and other jobs afterwards.

We have the following two design rationales. They have been widely used and evaluated as effective heuristics in job scheduling problems(e.g., [25], [11], [10]).

First, we apply the optimization in two phases: firstly with static optimizations, and secondly with runtime optimizations. In the static optimization phase, we grasp the optimization opportunities that can be exploited offline. For example, given the deadline of a job, we can assign tasks with individual deadlines so that tasks can be scheduled independently. In the runtime optimizations, we take advantage of the optimization opportunities can be only exploited in runtime. Example opportunities include virtual machine consolidation and task scheduling according to the available amount of green energy.

Second, our decision is made at the interval of a predefined *epoch* (15 minutes in our study). So we can leverage weather forecast that is usually quite accurate in such a short period, and further have the prediction on green energy supply. Also, within the epoch, our scheduling performs task assignment and migration by considering the cooling energy consumption.

With these two design rationales in mind, we develop a green-aware and cooling-aware scheduling algorithm *MinBrown* to minimize the brown energy consumption. Our scheduling algorithm has two major components: a scheduling framework and data center selection strategies. Our scheduling framework is general so that data center selection strategies for a given task can be easily incorporated into the framework. The awareness of green energy and cooling is considered in data center selection strategies.

Algorithm 1 MinBrown Scheduling Framework

1: /*Static optimizations*/
2: **if** a new job j is submitted **then**
3:     Perform deadline assignment on j;
4:     Put j's tasks into the scheduling queue;
5: /*Dynamic optimizations*/
6: **if** the slack time of a task $task$ becomes zero **then**
7:     /*select the data center with the minimum brown energy consumption; */
8:     $dc = GetDC(task, DC, 100\%)$;/*Algorithm 2*/
9:     schedule $task$ to dc;
10: **if** a new epoch begins **then**
11:     Predict the available green energy on each data center;
12:     Use longest remaining time first policy to pick tasks to migrate;
13:     Repeat the picking and migration process until there is no green energy available, or no task can be migrated;
14:     **while** green energy is available in any data center during the new epoch **do**
15:         Let the set of data centers DC_G;
16:         Use earliest latest starting time first policy to pick a task that is ready to run, $task_G$;
17:         /*select the data center with the minimum brown energy consumption and the usage of brown energy is lower than $b\%$ of the task energy consumption; */
18:         $dc = GetDC(task_G, DC_G, b\%)$; /*Algorithm 2*/
19:         **if** dc is not null **then**
20:             schedule $task_G$ to dc;
21: /*Below are local scheduler to each data center*/
22: **if** a task $task$ is scheduled on a dc **then**
23:     If no VMs are idle, wake up a physical machine;
24:     Return an idle VM to execute $task$;
25: **if** a task ends **then**
26:     A virtual machine becomes idle;
27:     Virtual machines are consolidated for further energy saving;
28: **if** a physical machine is idle for over a predefined time period (e.g., 3 minutes) **then**
29:     Set the machine to ACPI S3;

B. Scheduling Framework

Our scheduling framework (Algorithm 1) is general to different data center selection strategies, by allowing different implementation for *GetDC* in Lines 8 and 18. The detailed data center selection algorithms are presented in Section III-C. Lines 2–4 are static optimizations, and the rest lines are dynamic optimizations.

In the static optimization, we assign internal deadlines for each task of a job using existing Partial Critical Path Method [1]. After the deadline assignment, each task has its own deadline and the latest start time. Thus, each task can be scheduled independently.

In dynamic optimizations, the scheduler immediately dispatches the task without any slack, or periodically selects the tasks with slack to the suitable data center to exploit the opportunities of renewable energy. At the beginning of a new epoch, we first perform the prediction on the amount of green energy in the new epoch. The prediction method is orthogonal to this paper. There are a number of existing learning methods and models [10], [31], which has demonstrated high accuracy in short-term predictions.

With the prediction of renewable energy, we first consider migrating the currently running tasks until the green energy is expected to be used up or no task can be migrated. The task migration is implemented using virtual machine migration. Next, Lines 14–20 considers the scheduling of the tasks with slack in the earliest latest starting time first manner. Line 18 performs the data center selection according to our green-aware and cooling-aware strategy (Algorithm 2 in Section III-C).

Lines 22–30 are the functionalities for a local scheduler on each data center. It dynamically turns on/off the virtual machines as the task starts/completes. Furthermore, virtual machine consolidations are performed to improve the energy efficiency. If a physical machine is idle for a predefined period, the machine is set to ACPI S3 for energy saving. The techniques are classic energy saving techniques in virtualized environments [32], [21], [26]. We do not claim that they are a contribution of this paper. We briefly elaborate them here because energy saving techniques are still vital, even with renewable energy. Our scheduling framework embraces a series of classic energy saving techniques to reduce the usage of both brown and green energy. We refer readers to previous studies [32], [21], [26] for more details.

C. Data Center Selection Strategies

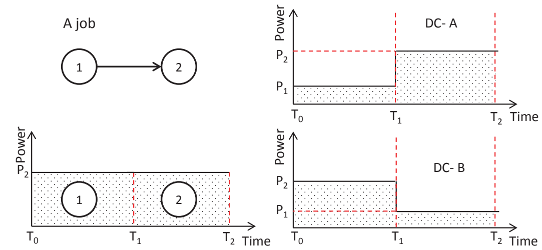
Before we present our data center selection strategy, we present the baseline algorithm without awareness of green energy or cooling – RR (Round Robin). RR is the simplest policy. If a data center is fully occupied, other data centers will be considered. This simple approach has two main problems in the green energy usage. First, without the awareness of green energy, it may schedule too much workload to a data center with little green energy, or schedule too little workload to fully utilize the available green energy. Second, without the awareness of cooling, its decision is oblivious to the outside temperate and cooling level upgrades.

Algorithm 2 MinBrown data center selection for Task $task$ from the data center candidate set DC: $GetDC(task, DC, b\%)$ ($b\%$ is the threshold brown energy ratio for selection)

1: $d = GetInitialDC(task, DC)$;
2: **if** the task does not have enough slack to transfer its intermediate input data to data centers other than d **then**
3:     return d;
4: **else**
5:     **if** d has sufficient green energy for $task$ and no cooling level is upgraded **then**
6:         return d;
7:     **for** each $d' \in DC/\{d\}$ **do**
8:         **if** d' has sufficient green energy for $task$ and no cooling level is upgraded **then**
9:             Migrate the intermediate input for $task$;
10:             return d';
11:     Calculate the data center with the minimum brown energy consumption (let it be d'');
12:     **if** d'' results in brown energy consumption ratio larger than $b\%$ **then**
13:         return null;
14:     **else**
15:         return d'';

We propose a cooling- and green-aware data selection strategy, as illustrated in Algorithm 2. The algorithm assigns a task to its suitable data center by considering the green energy usage of both cooling and computing facilities. At the first step of the algorithm (Line 1), we select the initial data center for the task. If the task is the root task, its input data has been replicated in all the data centers involved and the selected data center is the one with the largest available amount of green energy. Otherwise, it is the one resulting with the smallest network delay to transfer its intermediate input data from its parent tasks. Note, a task may have multiple parent tasks, and we select the one with the smallest data transfer overhead.

After assigning the initial data center, we check whether the task has sufficient slack time to transfer its intermediate input data to data centers other than the initial assignment. If not, the initial data center is selected for assigning the task. Otherwise, we need to consider how much brown energy



(a) A job and its energy requirement (b) Green energy supply at DC-A and DC-B

Figure 2. An example of a two-task job and two data centers. If the slack allows, task 1 runs on DC-B and task 2 runs on DC-A for maximizing the green energy usage.

the task would consume when it is scheduled to a data center (Lines 4–15). Lines 5–10 give the priority to the data center with sufficient green energy to serve the task without cooling level upgrade. The reason behind this optimization is that the cooling level upgrade usually causes a much more significant amount of energy consumption than a single task. If there is not sufficient green energy in all data centers or cooling level upgrade needs to be triggered, we calculate the data center that results in the minimum brown energy consumption (Line 11). If the brown energy ratio is smaller than a threshold value ($b\%$), we assign the task to that data center. Otherwise, the task can wait because it still has slack.

There is a common function in calculating the amount of brown energy consumption (e.g., Lines 5, 8 and 11). Consider a task starts at t_0 and completes at $t_0 + \delta t$. We denote the amount of renewable energy as a function of time ($g(t)$) and the energy consumption of the task to be ($w(t)$). Thus, the amount of brown energy consumption at time t by the task is given in Eq. 1.

$$\mathfrak{b}(t) = \begin{cases} w(t) - g(t) & \text{, if } w(t) > g(t), \\ 0 & \text{, otherwise.} \end{cases} \quad (1)$$

The total amount of brown energy by the task is $\int_{t_0}^{t_0+\delta t} \mathfrak{b}(t)$. After assigning the task, the available amount of green energy is $\int_{t_0}^{t_0+\delta t}(g(t) - (w(t) - \mathfrak{b}(t))$, where $(w(t) - \mathfrak{b}(t))$ represents the amount of green energy consumed by the task at time t.

Figure 2 illustrates a simple example of green-aware scheduling among two data centers. If we run the job on a single data center (e.g., DC-A), the amount of brown energy used is $(P_2 - P_1) \times (T_1 - T_0)$. In contrast, MinBrown schedules Task 1 on DC-B and Task 2 on DC-A so that both tasks are executed without any brown energy consumption (we ignore the task migration overhead in this example).

The algorithm needs to maintain the available amount of renewable energy in a data center as the tasks are assigned to the data center. Suppose the epoch size is N time units. The storage overhead is $O(N)$. The runtime overhead of making a decision for a task is $O(N^2)$. In the implementation, this runtime overhead is usually smaller than 1ms. This is ignorable for scientific and HPC workloads.

IV. Experimental Study

This section presents our experimental results.

A. Methodology

We use simulation to evaluate different workload scheduling algorithms. The simulator implements various issues in data center energy consumption including equipments and cooling.

Our simulated data centers are configured with the same setting as the previous studies [21], [12]. Each data center has the same Power Usage Efficiency (PUE), i.e., the total energy consumed by all facilities of the data center divided by the energy consumed by IT equipments. We assume an inter-data-center bandwidth of 464 Mbps. Each data center contains 480 servers, each of which has 4 cores and 4GB of memory and can host at most four single-core virtual machines. Our simulator accounts for the energy consumption at different states (idle, active and ACPI S3). Since the resource allocation is at the granularity of virtual machines, we use the number of cores to approximate the machine utilization. Suppose the number of allocated cores is c. We estimate the power consumption for a machine to be $P = base + \frac{c}{\#TotalCore}(peak - base)$, where the peak power ($peak$) and the base power ($base$) are set to 300W and 200W, respectively. Thus, $P = 200 + 25c$ in our simulation. The power consumption of ACPI S3 is 8.6W, and transiting into and out of S3 takes 7 seconds.

We use the same setting on the cooling facilities as in the previous study [21] (e.g., the data center inside temperate is set to be no more than 30 °C [3]) and adopt the same cooling strategy by considering the outside temperate. The cooling energy consumption is discrete in levels. For example, when the outside temperate is 25 °C and the data center utilization increases from 25% to a higher utilization, the cooling energy consumption increases by 55%. To avoid too frequent cooling level changes, we manually set a cooling level at least lasting for a certain time span, e.g., 5 minutes.

Each data center is associated with a solar farm. We simulate solar farms with different numbers of solar panels. The default size of a solar farm is 10,000 solar panels. Under the default setting, the peak green energy supply is equal to the peak energy consumption of each data center. We calculate the green energy production according to the parameter specification of BP-MSX 120 panels [28]. Basically, given instant irradiance and outside air temperature as input, the estimation model presented in the specification derives the amount of solar energy produced by a solar panel. The total amount of solar power is the power of each solar panel multiplied by the number of solar panels.

Workloads. We use the workload trace, the Parallel Workloads Archive [8]. Each trace entry consists of job id, submitted time, actual run-time, resource requirement (the number of cores etc). We tested multiple traces and observed similar results. For space limitation, we present the results for the trace LANL-O2K only. LANL-O2K contains approximately five months (November 1999 through April 2000) of jobs running on a 2048-node Origin 2000 cluster (Nirvana) at Los Alamos National Lab. We extract a random week from this trace (February 1-7, 2000) and use it as the workload for all experiments discussed below. This week-long trace contains 6799 jobs. There is a small portion of short running jobs (about 8% for less than 10 seconds), while there are also many long running jobs (10% for 10 hours or longer).

Solar Energy Trace. We use the real-world traces for solar energy from the Measurement and Instrumentation Data Center (MIDC) [30], because solar energy is widely available. We choose the trace from a random week (May 1–7, 2011), including irradiance and air temperatures. The time zone difference is a key measure on how much the green energy of two locations are complementary with each other. By default, we use traces from two stations (Loyola Marymount University Rotating Shadowband Radiometer and La Ola Lanai) to simulate two data centers DC-1 and DC-2. The two default data centers are located in Los Angels and Hong Kong (DC-1 and DC-2, respectively). Many companies like Google have their data centers in these two locations. Since we do not have meteorological data for DC-2, we simply use the data from La Ola Lanai by varying the time zone difference, because they are almost at the same latitude. The third and the fourth ones used are located in Virginia and Spain (using the same traces in DC-1 and DC-2, respectively). Note, we have to use this manual setting due to unavailable meteorological data sets in some locations, and this setting reflects the real-world data center deployment and is sufficient to demonstrate the effectiveness of our approach.

B. Results

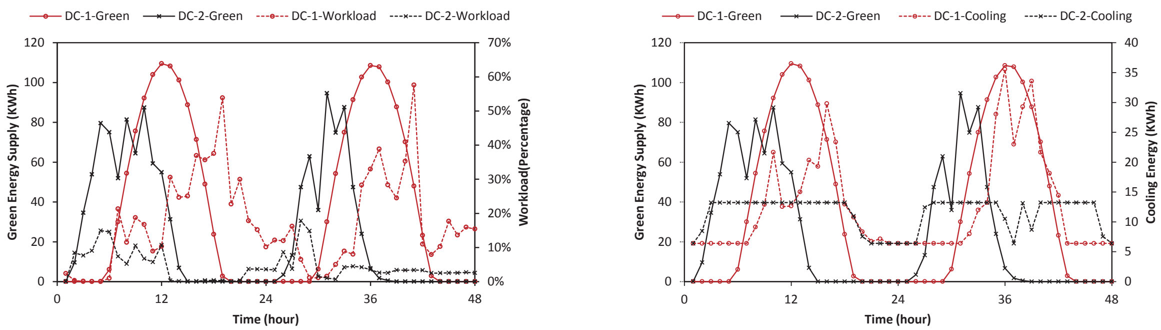
Recall that there are a series of parameters affecting the scheduling decision. For the space interests, we summarize the results of sensitivity studies in Table I, instead of figures, and comment some details on the findings below. To study the separate impact of individual optimization techniques, we consider three other baseline algorithms: 1) MBJ: The same as MinBrown except that the scheduling is conducted at the job level; 2) MB-NC: The same as MinBrown except that its decision does not consider cooling upgrade events; 3) MB-NG: The same as MinBrown except that it does not take green energy into account. Still, MB-NG considers task based scheduling and cooling upgrade events. We study the improvement as the ratio of brown energy reduction of MinBrown in comparison with other baseline algorithms.

In Table I, we use the following notations on the improvement as the parameter value varies from the left to the right (the default value is in bold): 1) "$a \rightarrow b\%$" means the improvement increases ($\nearrow$) or decreases ($\searrow$) from $a\%$

Table I

RESULTS FOR SENSITIVITY STUDIES ON MINBROWN. THE DEFAULT VALUE IS IN BOLD. THE IMPROVEMENT RESULT IS THE RATIO OF BROWN ENERGY REDUCTION OF MINBROWN IN COMPARISON WITH OTHER BASELINE ALGORITHMS.

Parameter	**Varying parameters**	**Improvement over RR**	**Improvement over MBJ**	**Improvement over MB-NC**	**Improvement over MB-NG**
Number of data centers	**2**, 3, 4	14 $\rightarrow$ 40% $\nearrow$	6 $\rightarrow$ 21% $\nearrow$	1 $\rightarrow$ 8% $\nearrow$	8 $\rightarrow$ 39% $\nearrow$
Time zone difference	2, 3, ..., **7**, ..., 12 hours	7 $\rightarrow$ 19% $\nearrow$	2 – 6%	2 $\rightarrow$ 1% $\searrow$	3 – 14%
Cooling downgrade delay	5, **10**, 20, and 30 minutes	14.0 – 14.2%	5 – 6%	1.0 – 1.4%	7 – 8%
PUE	1.1, 1.3, **1.5**	17 $\rightarrow$ 13% $\searrow$	7 $\rightarrow$ 5% $\searrow$	0 $\rightarrow$ 2% $\nearrow$	11 $\rightarrow$ 7% $\searrow$
Job slacks	1%, 5%, 10%, 15%, 20%, **25%**	13 $\rightarrow$ 14% $\nearrow$	2 $\rightarrow$ 6% $\nearrow$	2 $\rightarrow$ 1% $\searrow$	5 $\rightarrow$ 8% $\nearrow$
Migration overhead	**1**, 5, 20, and 30 minutes	11 – 15%	6 $\rightarrow$ 2% $\searrow$	1 $\rightarrow$ 4% $\nearrow$	8 $\rightarrow$ 4% $\searrow$
Brown threshold (b%)	**50%**, 60%, ..., 100%	14 – 15%	5.6 – 5.9%	1 – 2%	7.5 – 7.8%
Solar farm size	0.3, 0.5, 0.75, **1.0**, 1.5, 2.0 and 2.5	3 $\rightarrow$ 22% $\nearrow$	1 $\rightarrow$ 12% $\nearrow$	0 – 2%	3 $\rightarrow$ 12% $\nearrow$
Green scale ratio	(DC-1:DC-2) **1:1**, 1:2, 1:4, 1:6, 1:8	14 $\rightarrow$ 16% $\nearrow$	6 – 7%	0.9 – 1.4%	8 $\rightarrow$ 16% $\nearrow$
Green prediction error	**0%**, 5%, 10%, 15%	12 – 14%	6 $\rightarrow$ 4% $\searrow$	1 $\rightarrow$ 0% $\searrow$	8 $\rightarrow$ 5% $\searrow$
Outside temperature (DC-2)	-4, -2, **0**, 2 and 4 °C	16 $\rightarrow$ 11% $\searrow$	7 $\rightarrow$ 5% $\searrow$	1 – 2%	8 $\rightarrow$10% $\nearrow$
Summary	N.A.	2 – 40%	0 – 21%	0 – 8%	0 – 39%



(a) Workload (b) Cooling

Figure 3. A snapshot of cooling energy consumption and green energy supply in two days.

to $b\%$; 2) "$a - b\%$" means the improvement is between $a\%$ and $b\%$.

We first study the impact of parameters about data centers.

Number of Data Centers. The data centers are almost uniformly placed in different time zones. As the number of data centers increases, the improvement of MinBrown increases significantly. This means our approach is more suitable for cloud providers with multiple geographically distributed data centers. When the number of data centers is four, MinBrown outperforms MB-NC by 8%. Since PUE=1.5, the cooling energy consumption is around 33% of the total energy and our cooling aware strategy reduces around 49% of the total brown energy used in cooling.

Time Zone Difference. We intentionally shift the trace of DC-2 to illustrate the impact of time zone difference. While this may result in unrealistic location for data centers, the results are as expected that increasing the time zone makes two data centers have more opportunities in fully utilizing the green energy.

Cooling Downgrade Delay. The improvement of MinBrown is not sensitive to the cooling downgrade delay, as long as this delay is kept at reasonable value (5–30 minutes).

PUE. As the PUE increases, cooling plays a more significant role in energy consumption and our cooling aware optimization has larger impact. The improvement of MinBrown over MB-NC increases from less than 1% to over 2%.

We next study the impact of parameters about workloads.

Job Slacks. As the job slack increases, MinBrown has more opportunities in utilizing the green energy. So, the improvement over RR, MBJ and MB-NG increases.

Migration Overhead. We vary the migration overhead to simulate different amounts of data transfer during each job migration. Migration overhead affects the number of tasks migrated. Increased migration overhead reduces the benefit of workload migration. When the migration overhead increases from 1 minute to 30 minutes, the number of migrated tasks is reduced from 8% to 0.1% of the total number of tasks. Nevertheless, MinBrown still outperforms RR by 11–15% on brown energy reduction. Also, the total migration overhead is smaller than 1% of the total execution time, and thus the extra network energy consumption due to job migration is negligible.

Brown Threshold ($b\%$). The improvement of different b values is stable. This is because the green energy supply mostly can offer the energy consumption for many tasks. Varying b values only affects the decision of a small number of tasks at each epoch.

Finally, we study the impact of parameters about green energy supply and physical environments (outside temperature).

Solar Farm Size. Solar farm sizes in number of solar panels directly affect the amount of solar energy. As the solar farm size increases on both DC-1 and DC-2, the brown energy consumption of all approaches reduces. MinBrown has better green-aware scheduling, and the improvement over other approaches increases. However, it is not always beneficial to increase the solar farm size, due to the increased investment into the solar farm. Cloud providers should carefully provision the solar farm size for balancing the reduction in the electricity bill (in terms of brown energy from public grid) and the cost of running the solar farm. Since the goal of this study is on reducing the brown energy consumption, we leave this cost analysis as our future work.

Green Scale. We further consider the scenario of two data centers with different green energy supply. We fix the green energy supply of DC-1 and vary the ratio of green energy supply (DC-1:DC-2) from 1:1, 1:2, to 1:8. As the ratio decreases, the green supply of DC-2 increases. While MinBrown has larger improvement over other approaches, we find that a larger portion of green energy is wasted and should be used for other purposes.

Green Prediction. The prediction errors on the amount of green energy come from two major sources: weather forecast and the model on solar panel. Instead of relying on a specific prediction model, we use the real renewable energy as our prediction base, and explicitly study the impact of by adding prediction errors to the prediction base. Specifically, given a prediction error e and the real amount of solar energy g, the estimation is randomly distributed in $[g(1-e), g(1+e)]$. The results show that, if the prediction error is reasonable (less than 15% in our experiments), the improvement of MinBrown over other approaches has only a small degration.

Outside Temperature. We intentionally adjust the outside temperature of DC-2 by -4, -2, ..., 4°C. As the outside temperature of DC-2 becomes higher, more cooling energy is consumed for the same utilization. Thus, we observed more workloads are migrated to DC-1. The result improvement over MB-NC is 1–2%, which indicates that the scheduling algorithm should take the outside temperature and cooling into account even with green-aware optimizations.

After presenting the overall comparison, we conduct detailed studies in understanding the reduction of brown energy consumption of MinBrown for green-aware scheduling and cooling awareness. The cooling energy consumption and the workload in each data center is generally consistent with the trend of green energy supply, as illustrated in Figure 3. For example, at hours 1–6, DC-2 has more green energy and more workloads are performed on DC-2. Thus, the cooling energy consumption is also larger. After Hour 10, DC-1 has more green energy and the cooling energy consumption also increases on DC-1.

Finally, we study both brown and green energy consumption for completeness. Figure 4 shows the normalized brown, green and total energy consumption for the four approaches under default settings. The normalization base is the total energy consumption of RR. MinBrown consumes the smallest amount of brown energy as well as the lowest

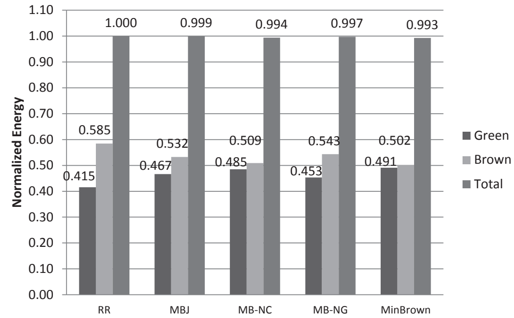


Figure 4. Normalized brown and green energy consumption.

total energy. In all other experiments, MinBrown sometimes consumes a larger amount of total energy (always less than 0.5%), but always consumes the least amount of brown energy. Also note that, the utilization of green energy by MinBrown is 53%, in contrast with RR with 46%.

C. Summary

Our studies demonstrate that the effectiveness of our optimizations in reducing brown energy consumption, and MinBrown outperforms all baseline algorithms under reasonable parameter ranges. All the three optimizations (i.e., green-awareness, cooling-awareness and task-based scheduling) contribute to the improved green energy usage. Among them, green-aware scheduling contributes the most, task-based scheduling the second and cooling-aware comes the last.

V. Related Work

We review the related work on utilizing green energy in data centers in two categories: studies for a single data center and for multiple data centers.

At the scale of a single data center, a number of studies [11], [10], [18], [2], [22] have been conducted to exploit renewable energy. Íñigo Goiri et al. leveraged renewable energy to handle scientific workloads [10]. They further integrated green awareness into Hadoop (namely GreenHadoop [11]). Krioukov et al. [18] advocated a supply-following computing paradigm for data intensive applications, and developed a green aware scheduling algorithm to maximize green usage while meeting the deadline of data processing jobs. Baris Aksanli et al. [2] developed green-aware scheduling for both on-line services and batch jobs in a single data center. Compared with this paper, these existing studies for a single data center have two major drawbacks: first, without considering the complementary effects on green energy supply among multiple data centers; second, these studies do not consider the discrete cooling granularity, which causes excessive cooling energy. There are also other proposals on architectural integrations into data centers (such as energy storage [27]). Our paper is complementary to these integrations.

At the scale of multiple data centers, many studies target at interactive Internet services [34], [24], [20], [19].

Zhang et al. [34] and Le et al. [19], [20] dynamically scheduled online services across multiple data centers to maximize green energy usage for Web hosting with cost budget constraints. Liu et al. [24] studied geographically load balancing policies. Since interactive services have little slack, most of these workload scheduling algorithms target at maximizing instant usage of green energy while satisfying online services QoS. Compared with this paper, they do not consider dynamic workload placement and migration. Moreover, they do not consider discrete cooling granularity, which is physically related to the outside temperature of geographically distributed data centers. Under the context of HPC jobs, this paper attempts to match the green energy supply and the workload by smartly scheduling the tasks within their slacks.

VI. Conclusions And Future Work

In this paper, we have studied the problem of minimizing brown energy consumption of HPC workloads for cloud providers that operate multiple geographically distributed data centers with renewable energy sources. Specifically, we propose a holistic workload scheduling method MinBrown considering green energy availability, cooling power and fine-grained scheduling. Our extensive studies have shown that our green-aware optimization, task-based scheduling and cooling-aware optimization achieve up to 39%, 21%, 8% reduction in the brown energy consumption, and combining all the three optimizations achieves 2–40% reduction over the load-balanced scheme. Our future work includes conducting price and cost analysis like our previous studies [33], [15], and extending MinBrown to other workloads.

Acknowledgement

The authors would like to thank anonymous reviewers for their valuable comments. This work is supported by an Inter-disciplinary Strategic Competitive Fund of Nanyang Technological University 2011 for "C3: Cloud-Assisted Green Computing at NTU Campus".

References

[1] S. Abrishami, et al. Cost-driven scheduling of grid workflows using partial critical paths. In *Proc. of Grid'10*, pages 81–88, 2010.

[2] B. Aksanli, et al. Utilizing green energy prediction to schedule mixed batch and service jobs in data centers. In *Proc. of HotPower'11*, pages 5:1–5:5, 2011.

[3] ASHREA. Environmental guidelines for datacom equipment. In *American Society of Heating, Refrigeration, and Air-Conditioning Engineers.*, pages 1–11, 2008.

[4] L. A. Barroso and U. Hölzle. The case for energy-proportional computing. *Computer*, 40(12):33–37, Dec. 2007.

[5] R. Chen, X. Weng, B. He, M. Yang, B. Choi, and X. Li. Improving large graph processing on partitioned graphs in the cloud. In *Proc. of ACM SoCC'12*, 2012.

[6] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, Jan. 2008.

[7] X. Fan, et al. Power provisioning for a warehouse-sized computer. In *Proc. of ISCA'07*, pages 13–23, 2007.

[8] D. Feitelson. Parallel Workloads Archive. http://www.cs.huji.ac.il/labs/parallel/workload/.

[9] K. Gao. U.S.A. and China solar pv market and ppa solar project development. In *Proc. of ISETC'11*, 2011.

[10] I. Goiri, et al. GreenSlot: Scheduling energy consumption in green datacenters. In *Proc. of SC'11*, pages 20:1–20:11, 2011.

[11] I. Goiri, et al. GreenHadoop: Leveraging green energy in data-processing frameworks. In *Proc. of EuroSys'12*, pages 57–70, 2012.

[12] A. Greenberg, et al. The cost of a cloud: research problems in data center networks. *SIGCOMM Comput. Commun. Rev.*, 39(1):68–73, December 2008.

[13] B. He, M. Yang, Z. Guo, R. Chen, W. Lin, B. Su, H. Wang, and L. Zhou. Wave computing in the cloud. In *HotOS*, pages 1–5, 2010.

[14] B. He, M. Yang, Z. Guo, R. Chen, B. Su, W. Lin, and L. Zhou. Comet: batched stream processing for data intensive distributed computing. In *Proc. of ACM SoCC'10*, pages 63–74, 2010.

[15] S. Ibrahim, B. He, and H. Jin. Towards pay-as-you-consume cloud computing. In *Proc. of SCC'11*, pages 370–377, 2011.

[16] S. Ibrahim, H. Jin, L. Lu, B. He, G. Antoniu, and S. Wu. Maestro: Replica-aware map scheduling for mapreduce. In *Proc. of CCGRID'12*, pages 435–442, 2012.

[17] J. Koomey. Worldwide electricity used in data centers. *Environmental Research Letters*, pages 3:1–3:8, 2008.

[18] A. Krioukov, et al. Integrating renewable energy using data analytic systems: Challenges and opportunities. *IEEE Data Engineering Bulletin*, pages 1–9, 2011.

[19] K. Le, et al. Cost- and energy-aware load distribution across data centers. In *Proc. of HotPower'09*, pages 1–5, 2009.

[20] K. Le, et al. Capping the brown energy consumption of internet services at low cost. In *Proc. of IGCC'10*, pages 3–14, 2010.

[21] K. Le, et al. Reducing electricity cost through virtual machine placement in high performance computing clouds. In *Proc. of SC'11*, pages 22:1–22:12, 2011.

[22] C. Li, et al. Characterizing and analyzing renewable energy driven data centers. In *Proc. of SIGMETRICS'11*, pages 323–324, 2011.

[23] C. Li, et al. iswitch: Coordinating and optimizing renewable energy powered server clusters. In *Proc. of ISCA'12*, pages 512–523, 2012.

[24] Z. Liu, et al. Greening geographical load balancing. In *Proc. of SIGMETRICS'11*, pages 233–244, 2011.

[25] M. Mao and M. Humphrey. Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In *Proc. of SC'11*, pages 1–12, 2011.

[26] M. R. Marty, et al. Virtual hierarchies to support server consolidation. In *Proc. of ISCA'07*, pages 46–56, 2007.

[27] G. Sriram, et al. Benefits and limitations of tapping into stored energy for datacenters. In *Proc. of ISCA'11*, pages 341–352, 2011.

[28] BP MSX 120 Solar Module. http://pdf.directindustry.com/pdf/bp-solar/bp-msx-120-solar-module/15873-68158.html.

[29] Data Center Knowledge. How google's data centers use renewable energy., 2012.

[30] Measurement and Instrumentation Data Center. http://www.nrel.gov/midc/.

[31] N. Sharma, et al. Cloudy computing: Leveraging weather forecasts in energy harvesting sensor systems. In *Proc. of SECON'10*, pages 1–9, 2010.

[32] T. Wood, et al. Cloudnet: Dynamic pooling of cloud resources by live wan migration of virtual machines. In *Proc. of VEE'11*, pages 121–132, 2011.

[33] H. Wang, Q. Jing, R. Chen, B. He, Z. Qian, and L. Zhou. Distributed systems meet economics: pricing in the cloud. In *Proc. of HotCloud'10*, pages 6–6, 2010.

[34] Y. Zhang, et al. GreenWare: Greening cloud-scale data centers to maximize the use of renewable energy. In *Proc. of Middleware'11*, pages 143–164, 2011.