Green Databases Through Integration of Renewable Energy

Cheng Chen[2,1], Bingsheng He,
Xueyan Tang, Changbing Chen
[1]Nanyang Technological University, Singapore

Yubao Liu
[2]Sun Yat-sen University, China

ABSTRACT

Recently, a lot of energy efficient techniques have been developed to reduce the usage of the carbon intensive energy (*brown* energy) of databases. There is a new opportunity to reduce the brown energy usage: renewable energy (*green* energy) has been used to at least partially power computer systems. The key challenge of exploiting green energy sources is that they are variable and intermittent. So far, there has been little work on integrating renewable energy into a database system. This paper attempts to bridge this gap with ReinDB (Renewable Energy Integrated Database). The design goal of ReinDB is to minimize the brown energy consumption on a database server with both green and brown energy supplies. Specifically, we develop the green supply driven execution paradigm and adaptive power management techniques to adapt to green energy supply. We further propose green aware optimizations to improve the efficiency of both brown and green energy usage. Preliminary results demonstrate the effectiveness of ReinDB in reducing the brown energy usage.

1. INTRODUCTION

Energy efficient techniques have attracted a lot of research efforts to reduce the carbon footprint of databases. Most techniques (e.g., [12, 20, 21, 33, 19]) target at reducing the energy consumption in order to reduce the environmental impact of traditional energy sources (*brown* energy). Recently, with increasing adoptions of renewable energy supply techniques (such as solar panels) [8], some computer systems have been powered at least partially with renewable (or *green*) energy. For example, Samsung releases a solar-powered mini laptop NC125S that significantly extends the battery lifetime with solar panels. A number of data centers have used green energy near their locations [30]. Green energy usage creates new opportunities for reducing brown energy usage of databases. This paper aims at minimizing the brown energy usage of a database server powered by both brown and green energy sources.

The key challenge of integrating green energy into computer systems is that green energy sources are variable and intermittent due to daily/seasonal effects. The mismatch between workloads and green energy supply challenges the utilization of green energy and thus potentially causes significant brown energy usage. There are a number of previous studies [10, 9, 17, 36, 25] on addressing the mismatch in the context of data centers. However, those techniques are specific to data centers, and are not directly applicable to an individual database server or data management systems. On the other hand, existing energy saving database techniques [12, 20, 21, 33, 19] are green-oblivious. While they can reduce the total energy consumption of database servers, they are not aware of the mismatch between green supply and workloads, and may cause excessive usage of brown energy.

To bridge this gap, we propose ReinDB (Renewable Energy Integrated Database) with the goal of minimizing the brown energy consumption of databases. ReinDB runs on a database server powered by both green and brown energy sources. Workloads are submitted with their predefined slack times. The slack time allows us to adopt the green supply driven execution paradigm previously proposed for data centers [17]. Specifically, we implement the supply-driven execution (SDE) on a database server with power management (voltage/frequency scaling on CPU) and workload scheduling to avoid slack violations. Due to non-energy-proportional feature of modern servers [3, 33], each joule of energy supply can complete different amounts of workload. We further propose green aware optimizations to improve the efficiency of both brown and green energy usage. In those optimizations, ReinDB dynamically shifts the workload to operate the database server at the most energy efficient point. Battery is also used to assist the workload shifting.

As a proof-of-concept study, we have developed a simulator to evaluate ReinDB. The simulator takes the solar energy trace and workload trace as input, and replays the traces according to different execution strategies on the simulated battery and database server. Our preliminary results demonstrate the effectiveness of our optimizations in reducing the brown energy usage (up to 57% and 53% reduction compared with green-oblivious and basic green-aware approaches [17], respectively). Moreover, it achieves almost the same amount of total energy consumption as other approaches.

The initial design and evaluation of ReinDB has shed light on future database systems with the integration of renewable energy. We further outline a number of research directions on green databases through integrating renewable energy. We hope that this paper will bring awareness to the database systems community about the important opportunities of renewable energy for research on green databases, particularly towards zero-emission databases.

Organization. The remainder of this paper is organized as follows. Section 2 formally defines the problem. Section 3 presents our detailed design of ReinDB, followed by the preliminary experimental results in Section 4. We outline a number of

6th Biennial Conference on Innovative Data Systems Research (CIDR '13) January 6-9, 2013, Asilomar, California, USA.

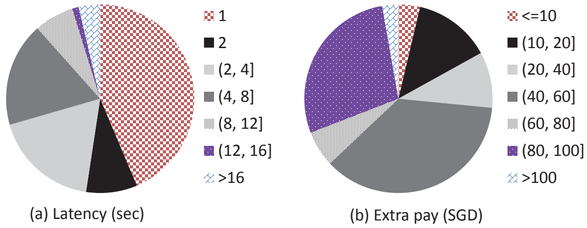


Figure 1: The response results of Q1 and Q2 in the user survey (with a population of 355).

research directions towards green databases with renewable energy in Section 5. We review the related work in Section 6 and conclude this paper in Section 7.

2. PROBLEM DEFINITION

In this section, we present our motivation of integrating renewable energy into database systems, followed by our system overview.

2.1 Motivation

The energy consumption behavior of users significantly affects the design of green computing systems. The effectiveness of ReinDB relies on two major factors: slack (how much latency users can tolerate to complete a task with green techniques) and economics (how much extra pay users can tolerate on buying the green solution, e.g., new hardware). The slack time allows advanced scheduling mechanisms to better exploit green energy supply, and the economics issue justifies the new hardware for supporting green energy.

To understand the factors of slack and economics, we collaborate with a psychologist and conduct a user study. The survey is a part of the multi-disciplinary research project aiming at reducing the energy consumption of computing systems. The following two questions are relevant to this paper: (Q1) If it currently takes about 1 second on average to complete a search in Google, how long are you willing to wait additionally for a green platform? (Q2) Imagine there is a new green computer system design using solar energy (sunlight) in addition to electricity. If a regular system costs about 100 SGD, how much are you willing to pay for the solar powered system?

We have interviewed 355 persons (230 from the campus of Nanyang Technological University and 125 from Singapore Changi airport). They are with different nationalities and cultures. The results are shown in Figure 1. Overall, users can tolerate quite significant amount of latency and extra pay for green solutions. For Q1, the average tolerable slack by users is 4.5 sec, indicating an average slack of over 4.5 times of the task execution time. Over 3.7% of them allow slacks of more than 16 times of the task execution time. For Q2, the average extra pay is 64.1 SGD, which is 64.1% of the original price. Almost 3/4 of the users are willing to pay 40–100 SGD more. That motivates us to develop a green system to minimize the brown usage with a reasonable increase in the latency and hardware budget.

2.2 System Overview

We have witnessed different scenarios of integrating renewable energy into computing systems: a single machine (e.g., Samsung NC125S), and data centers [30]. As a start, we consider the former scenario of integrating renewable energy into a single machine.

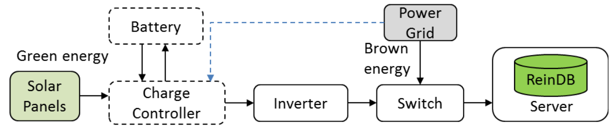


Figure 2: ReinDB runs on solar powered database server.

The proposed system design is applicable to individual scale-out architectures in the data center.

Database server. ReinDB is designed to run on a server powered by both green and brown energy sources. Its architectural design is illustrated in Figure 2. The system comprises a server, an inverter, batteries and a switch. Batteries are optional, because they cause severe environmental issues. Some servers are equipped with batteries for emergency. As the input voltage from the renewable energy source varies dynamically, the charge controller regulates the charging and prevents any over charging. The charge controller offers the monitoring on the charging/discharging operations, including voltage, current and power. With the charge controller, the surplus green energy is automatically charged into the battery.

With the penetration of renewable energy into public grids, electricity from public grids becomes partially brown. Nevertheless, our proposed approach can still be applicable by considering the portion of brown energy in the public grid. A smart switch is used to be connected with both energy sources. 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 and/or the battery supply, it immediately draws power from brown sources.

ReinDB relies on server power consumption characteristics and the battery model to make workload scheduling decisions.

Server power consumption. Previous models [19, 15] have estimated the total power consumption of a computer to be the total power consumption of hardware components (such as CPU, DRAM and disks, and other components). The power consumption of a hardware component mainly depends on the hardware setting (e.g., the maximum frequency for CPU) and the runtime utilization. The hardware setting can be adjusted with common techniques: 1) DVFS (Dynamic voltage/frequency scaling) on CPU and DRAM; 2) multi-speed hard disk. Some hardware setting may not be adjustable at runtime. In our study, we cannot adjust the hardware setting for DRAM and disks at runtime, and thus we use DVFS for CPU only.

Modern servers are far from energy-proportional [3, 33], because the idle power can constitute a major portion of the total power consumption. For example, it can contribute to more than 50% of the energy consumption of a database server [33]. ACPI S3 is commonly used to reduce the idle consumption. The disadvantage of using S3 is that it causes a latency for the server to resume responding. On our server, it takes about 2.5 seconds to go into S3 state, and 4.5 seconds from S3 back to normal state.

Battery model. The charging/discharging behavior of a battery is complex, which attracts a lot of research attention [32, 28]. Charging causes inherent energy loss, and the loss rate usually increases for most batteries along the charging cycles. For discharging, the battery cannot be fully discharged, because the inverter normally disconnects the battery supply when the voltage is lower than a threshold.

We model the battery with three parameters $< C_p, e(c), loss >$, where C_p is the capacity of the battery (Watt·hour, or Wh), $e(c)$ is the charging power of the battery at the capacity c ($0 \leq c \leq C_p$) and *loss* is the average energy loss of charging the battery.

When the battery is fully charged, the amount of energy loss for a complete discharging is $C_p \times loss$. We use $e(c)$ to capture different speeds of the battery charging. At the capacity c, we estimate the energy storage increment as $e(c)$. If the power supply is higher than $e(c)$, the surplus power cannot be charged into the battery. This model is applicable to different kinds of batteries (e.g., Lead-Acid [32] and Lithium-Ion [28]), by calibrating different curves of $e(c)$. While various complex models have been developed for different kinds of batteries [24], our model is simple and useful in guiding our calibration.

Workload. We study database workloads (either online transactions or analytical queries). ReinDB allows users to specify QoS: each query has a specified deadline to define its slack. We note that these deadlines are "soft deadlines" for quality-of-service purposes, unlike "hard deadline" as in hard real-time systems.

Optimization goal. Given a workload with predefined slacks, ReinDB dynamically schedules the workload and adjusts the hardware setting according to the green energy supply. The optimization goal is to minimize the total amount of brown energy usage, given the constraint that all queries in the workload are completed within their predefined deadlines.

3. GREEN DATABASE: REINDB

In this section, we start with an overall design of ReinDB. Then, we give the detailed design of key components in ReinDB.

3.1 Overall Design

Supply driven execution (SDE) is widely used in scheduling the workload in the data centers with renewable energy sources [17, 10, 9]. The core idea of those approaches is to align power consumption of the workload with the green supply. The scheduling is usually based on the heuristic of scheduling jobs in the increasing order of their slacks.

Due to the difference between data centers and database servers, there are a number of technical issues in applying SDE to a database server. The first issue is how to control the power consumption of a database server in order to align workload with the green supply. The complexity comes from the interwined hardware components (CPU/DRAM/Disk), which requires advanced modeling on their performance and power characteristics. Second, naive SDE may defer too many workloads. Due to the limited processing capability of a database server, that may cause aggregated performance bottleneck and excessive deadline violations. Third, since a database server is not energy-proportional, we should carefully use brown and green energy in their most efficient way. Fourth, battery adds a new dimension for optimizing the green energy usage. Most previous studies [17, 10, 9] have not considered the battery usage.

ReinDB adopts SDE to the context of a database server. ReinDB considers the power and performance for different hardware settings on power management, and periodically adjusts hardware settings according to workload and green supply (the period is called *slot*). In order to resolve the aggregated performance bottleneck, we leverage workload prediction and develop a multi-queue algorithm to schedule the queries according to the processing capability. We further propose optimizations to improve the efficiency of brown and green energy usage. We shift the workloads within their slacks to the points with higher energy efficiency. We develop two versions of energy efficient workload shifting: *brown-aware* and *battery-aware* for servers without and with battery. The optimizations (including multi-queue scheduling and workload shifting) are periodically performed to optimize the SDE schedule (the period is defined as *epoch*). An epoch is designed to be larger than a slot. In our design, an epoch consists of multiple slots. At the beginning of each epoch, we need to perform predictions on workloads and the green supply (e.g., through weather forecast). The overall design is described in Algorithm 1. Lines 1–7 generate an execution plan for the epoch. The execution plan specifies the amount of workload allocated and the hardware setting for each slot.

Algorithm 1 Overall design of ReinDB

```
1: if Current time is the beginning of an epoch then
2:     Conduct predictions on the workload and the green energy
       supply;
3:     Generate the supply driven execution plan, P; /*Section 3.2*/
4:     if The server does not have battery then
5:         Perform brown-aware workload shifting on P;/*Section 3.3*/
6:     else
7:         Perform battery-aware workload shifting on P;/*Section 3.4*/
8: Schedule workload execution according to P;
```

3.2 Supply driven execution

When we implement SDE on the database server, we develop two mechanisms for each slot: workload scheduling to meet workload deadlines and dynamic configurations on hardware setting.

At the beginning of an epoch, we perform the prediction on the workload (including query types and arrival rates etc) and the green supply for the epoch. Most green sources such as solar and wind depend on weather. The prediction method is orthogonal to this paper. There are a number of existing weather learning methods and models [14, 9] and workload prediction [7], which have demonstrated high accuracy in short-term predictions. Instead of relying on a specific prediction model, this paper uses the real value as our prediction base, and explicitly studies the impact of prediction accuracy by adding prediction errors to the prediction base in Section 4.

Achieving Workload Deadlines. We assume that all the queries in the queues can be completed with the demand-driven (DD) execution (i.e., queries are scheduled for executions once they are submitted). Otherwise, that means the system is overloaded. To amend the potential bottleneck of SDE execution, we leverage workload prediction and make sure that the queries are deferred only if all the queries can be finished by their deadlines by running the database server at its full speed.

We develop a multi-queue structure to maintain the queries with different slacks. Each epoch consists of N slots and the slot length t_s. Suppose there are N queues, q_0, q_1, ..., q_{N-1}, where q_i ($0 \leq i < N$) maintains all the queries with the slack time between $(i-1) \cdot t_s$ and $i \cdot t_s$. As the time goes by, the slack of existing queries decreases, and newly arriving queries are added to their corresponding queues. In the implementation, we maintain the queue structures in a circular queue. We use a head pointer that always points to the queries with slack of zero. At the ith slot, the slack time of the queries in q_i at the beginning of the epoch has decreased to [0, t_s]. A separate queue q' stores all the queries with the slack time beyond the current epoch. At the beginning of each epoch, the queries with the slack time smaller than the epoch size are distributed to corresponding q_is. Thus, the total workload for an epoch includes the queries from q' of the previous epoch and the queries that arrive during the epoch with deadlines before the end of the epoch. *Adjustment on* q_0, q_1, ..., q_{N-1} *is only performed at* *the beginning of a slot, and adjustment on q' is performed at the beginning of an epoch.*

Within a queue, the scheduling is performed in the shortest slack time first manner. Within the slot, queries are scheduled for processing at the same speed.

We generate the execution plan of an epoch so that each slot can finish all the queries allocated to it. For simplicity of presentation, we assume that each query consumes the same amount of computational resource and power. Extending our algorithm to handle queries with different computation demands is straightforward. Specifically, at the beginning of a new epoch, we investigate all q_i ($0 \le i < N$) and the predicted future workload of the coming N slots, and decide the amount of workload to be performed in the current slot so that the q_0 of each of the N coming slots does not exceed the processing capability of the database server. Given the query rate and their slack specification, we can easily calculate the number of queries that are required to be completed in each slot (denoted as m_i, $0 \le i < N$). Suppose a query is expected to be submitted after t seconds with a slack of k, it will be added to $q_{\frac{t+k}{t_s}}$ of the current slot. After obtaining m_i ($0 \le i < N$), we consider the scheduling from m_{N-1} to m_0. If m_i ($i > 0$) exceeds the full processing capability (cap), we need to move $m_i - cap$ queries to the ($i-1$)th slot, i.e., m_{i-1} is increased by $m_i - cap$. The moved queries are submitted at ($i-1$)th slot or even earlier. Due to the assumption that all the queries can be completed with the demand-driven (DD) execution, we are always able to find such queries to move. The process is done until m_0 is updated. In the extreme case where the server is always fully occupied, after scheduling, our approach degrades to the DD execution.

Dynamic configuration on hardware setting. The hardware setting is fixed within a slot. At any time within the slot, the amount of brown energy consumption depends on the relative magnitude of the power consumption of processing the queries (p_0) and the green power (g_0). If $p_0 > g_0$, we need extra energy ($p_0 - g_0$) from the brown supply. Otherwise, more queries can be scheduled until the green power is fully utilized. To maximize the energy efficiency, we formulate these two cases into a pair of dual problems. 1) $p_0 > g_0$: Minimizing the brown usage is to minimize the energy consumption p_0 for completing the workload. This problem is formulated as finding the optimal hardware setting so that the energy consumption is minimized for the given workload. The traditional energy saving techniques are relevant for this problem [19]. 2) For the latter case $p_0 < g_0$, our problem is to find the optimal hardware setting so that the amount of workload done is maximized, subject to a energy consumption budget.

We adopt the framework [19], and use learning methods including linear expression and neural network to develop the power-performance profile of executing a workload on a database server. For training, we run the target workload with different configurations (e.g., the number of clients in the TPC-C benchmarks) and with all different hardware settings (i.e., DVFS for the CPU in this paper). For each training execution, we calibrate the power and throughput. All the calibrated results essentially construct a mapping: (hardware setting, workload)–> (power, throughput). We filter the entries with low energy efficiency, which has a higher power with the same or lower throughput or has a lower throughput with the same or higher power. After filtering, for a given power budget, the mapping can give us the highest throughput. We further use learning methods to develop a full power-performance profile to solve the problem 2). Similarly, we can develop performance-power profile to solve problem 1).

In order to take advantage of ACPI S3, we must consider a tradeoff. ACPI S3 is useful only when the idle period is sufficiently long. Otherwise, it will cause energy loss during the state transition. The problem is similar to the power management of state transitions for DRAM. We adopt a simple approach developed by Wu et al. [35]. The basic idea is that the server is demoted to S3 only when it has been idle for Δ time. We maintain the histogram of idle periods in the previous epoch and use it to guide our decision of Δ for the current epoch. Let t be the length of an idle period. If $t \le \Delta$, there is no state transition and the energy consumption for the idle period is denoted as a function $\mathcal{B}_{short}(t) = P_{idle} \times t$, where P_{idle} is the idle power. Otherwise, the energy consumption is $\mathcal{B}_{long}(\Delta, t) = P_{idle} \times \Delta + P_{S3} \times (t - \Delta) + P_{loss}$, where P_{loss} is the penalty of power consumption of transiting from/to S3 and P_{S3} is the energy consumption of S3 state.

During the execution, we maintain the histogram of the idle periods in each epoch. Given the histogram $Hist[i]$, $i = 0, 1, ..., T$ (T is the epoch length in seconds), each $Hist[i]$ means there are $Hist[i]$ idle periods with length i seconds. We can calculate the total energy consumption for all the idle periods, as $E(\Delta)$ in Eq. (1). Our goal is to get the suitable demotion time Δ so that $E(\Delta)$ is minimized. We note that $E(\Delta)$ is neither concave nor monotonic. Therefore, we have to iterate all the possible values Δ=0, 1, ..., T, and find the suitable Δ.

$$E(\Delta) = \sum_{i=1}^{\Delta}(\mathcal{B}_{short}(i) \cdot Hist[i]) + \sum_{i=\Delta+1}^{T}(\mathcal{B}_{long}(\Delta, i) \cdot Hist[i]) \quad (1)$$

3.3 Brown-Aware Workload Shifting

SDE has aligned the workload according to green supply. However, due to a server's non-proportional energy consumption, each joule of energy supply can complete different amounts of workload. Since brown energy is used once the power usage is smaller than the green power, brown energy may not always be used at their optimal efficiency.

We propose a workload shifting method to improve the energy efficiency of brown energy usage of the execution plan given by SDE. At the beginning of a new slot, we consider shifting the workload of the coming N slots (denoted as s_0, s_1, ..., s_{N-1}) within an epoch. Specifically, given the workload schedule in s_i ($0 \le i < N$), we examine whether a slot needs brown energy in the order from s_0 to s_{N-1}. Suppose slot s_i ($0 \le i < N$) needs to use brown battery, we consider whether there are some workloads that are submitted before s_{i+1} and have slack at the slot s_j ($i < j < N$) to s_i. The queries are moved only when we find that it can reduce the amount of brown energy. Thus, we consider workload shifting between two slots with brown energy usage. We compare the energy efficiency of one joule of energy in s_i and s_j ($i < j < N$). Suppose there are k feasible slots that have brown energy usage and have a lower energy efficiency than s_i (denoted as s_{j_1}, ...,s_{j_k}, $i < j_z < N$, $z \in [1, k]$). We shift the workload from those k slots to s_i. The priority is given to the slot with the highest brown energy consumption gain. If s_i reaches its peak power or there is no brown energy reduction on the shifting, we consider other slots. We repeat this process until all queries have no slacks, or all queries are moved, or there is no brown energy reduction by workload shifting. After workload shifting, ReinDB takes advantage of ACPI S3 for further energy saving.

3.4 Battery-Aware Workload Shifting

So far, SDE and brown-aware workload shifting in ReinDB assume the battery is only for emergency use (e.g., when external energy sources are unavailable). The basic functionality of the battery is to store the surplus green energy to lengthen the discharging lifetime. However, if we allow battery to charge/discharge dynamically during workload execution of ReinDB, the battery essentially becomes a new factor for reducing the brown energy usage.

We can use the battery in a simple manner. When the demand is higher than the green supply, battery is first discharged. If the battery does not provide sufficient supply, the brown energy is used. This is the baseline approach of utilizing the capability of dynamic charging/discharging of the battery.

We can go beyond the baseline approach of using the battery by optimizing the energy efficiency of green supply. Consider the scenario that the green power is just slightly above the idle power. SDE will schedule a very small amount of workload to execute. This execution has relatively low energy efficiency, because of a server's non-proportional energy consumption. In fact, we can shift the workload to other times, even with slight increases on brown energy. Then, the green energy can be stored into the battery to reduce more brown energy usage in the future.

We develop our battery-aware optimization as a further optimization for the execution plan given by SDE. Our optimization is periodically performed on the schedule at the beginning of each epoch. Our algorithm considers the tradeoff of scheduling the workload. On the one hand, if we schedule the queries from the less energy efficient slot s_l to the more energy efficient slot s_h, we can use less green energy to finish those queries. Thus, the surplus green energy in s_l can be stored in the battery. On the other hand, the battery has a limited capacity and causes energy loss. Additionally, the charging speed varies along the charging cycles. We must take those issues into consideration so that the brown energy consumption is reduced.

At the beginning of a new epoch (consisting of N slots s_0, s_1, ..., s_{N-1} in time order), we consider the workload shifting from slot s_0 to s_{N-1}. The workload shifting algorithm is similar to that in brown-aware workload shifting, except that the battery-aware optimization considers all the slots and the brown-aware optimization considers only the slots with brown energy usage. Since the battery-aware optimization considers all the slots, it is more likely to create longer idle periods to take advantage of ACPI S3 state.

4. PRELIMINARY RESULTS

We have developed a simulator to evaluate the potential of ReinDB. Our simulation is developed based on calibrations in order to capture the key performance/power characteristics of real hardware.

4.1 Experimental Setup

Our simulator takes the trace from solar energy and workload as input, and outputs the performance and energy usage. We simulate the charging/discharging life cycles of battery and the power-performance profile of the server. The configurations are calibrated from real hardware, i.e., the battery (Fujitsu lithium ion battery) and the server (HP Z420 Workstation with an Intel Xeon E5-1650 CPU, 16GB RAM, and an OCZ Vertex4 256GB SSD). The peak, idle and S3 power consumption of the server is 280W, 84W and 6.5W, respectively. The server represents a main-stream servers with multi-core CPUs, dozens of GB RAM and SSDs. Since most data centers utilize commodity servers as scale-out architectures, the experimental results are relevant to the integration of renewable energy into data centers.

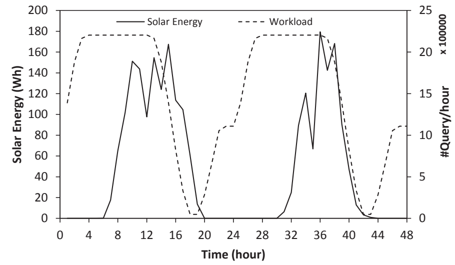


Figure 3: A snapshot of solar trace obtained from MIDC, and the sample workload simulating Google's search workload [26].

The CPU is a major energy consumer in modern computers [3, 33]. We study the energy profile of CPU with different frequencies (via DVFS) and different CPU utilizations. On the tested CPU, we have 16 levels for DVFS from 1.2 GHz to 3.2 GHz. We consider the utilization from 0% to 100% by varying amounts of workloads being executed on the database system.

Solar energy trace. We use the real-world traces for solar energy from Measurement and Instrumentation Data Center (MIDC) [31], because solar energy is widely available. We choose the trace of a random week (May 1–7, 2011), including irradiance every minute. Figure 3 illustrates a snapshot of the first two days. The solar energy supply is intermittent. We consider different solar power scales to evaluate the impact of different numbers of solar panels. By default, the peak solar power supply is provisioned to the peak power usage of the database server.

Workload trace. We experimentally obtain the power-performance and performance-power profiles of both transactional and analytical workloads on the database server. The database is PostegreSQL 9.1.3. We exercise different settings for workloads and CPU on the tested database server. For transactional workloads, we run TPC-C with 100 warehouses. We set the buffer size to be 6GB so that most data resides in the main memory. We vary the number of clients from 1, 2, ... to 32. For analytical workloads, we consider TPC-H Q5, and vary the number of Q5 instances running at the same time from 1 to 4. The scale factor is 50 and the buffer size is 2GB. We reset the database for running each setting. Both benchmark workloads are run with Database Test Suite [5].

We synthesize two kinds of workloads with different temporal patterns: 1) stable: the query rate is stable. We have four levels for different utilizations: S1, S2, S3 and S4 (around 10%, 20%, 40% and 80% of the peak performance of the server respectively). The query rate per second is 77, 154, 308 and 616 for TPC-C accordingly, and the query rates per hour are 1, 2, 4 and 8 for TPC-H Q5 accordingly. 2) dynamic: the query rate is dynamic. We simulate the query rate of TPC-C according to the daily pattern in Google search [26], as illustrated in Figure 3. We denote this workload as D1. The peak query rate per second for D1 is 616. In all those workloads, we use the slack setting obtained from the survey (ref. Section 2). We set the baseline response time for a transaction to be 1 and 363 seconds for TPC-C and TPC-H Q5 respectively.

Comparisons. We have adopted/implemented previous approaches for comparison: (1) "GO": the green oblivious normal execution. Queries are scheduled for executions without delay. (2)"SDE" [17]: the SDE execution in ReinDB with energy-efficient hardware setting and with achieving the deadline requirement.

To study the impact of individual optimization techniques, we consider the following ReinDB variants: 1) "ReinDB(WS)": SDE with brown-aware workload shifting; 2) "ReinDB(B-Basic)": SDE with the baseline battery usage; 3) "ReinDB(B-WS)": SDE

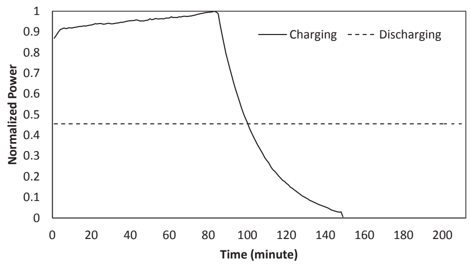


Figure 4: Charging/discharging power of the battery.

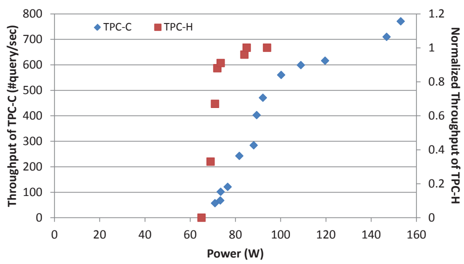


Figure 5: Power-performance profile of running TPC-C and TPC-H Q5.

with battery assisted workload shifting. The default epoch sizes are 15 minutes and 3 hours for TPC-C and TPC-H workloads, respectively. The slot size is 10 seconds.

4.2 Calibrations

We have calibrated the battery and the server. The calibration results are used as input to our simulator.

Battery. Figure 4 shows the charging/discharging power on the battery. The power is normalized to the peak charging power. The power is measured using the power meter (Refer to Table 1 in Appendix A). We use brown energy to charge the battery, and use a laptop that runs with consistent workloads to draw power from the battery. Thus, the discharging power is stable. In contrast, the charging power shows an interesting curve. At the beginning, the charging power is high around 0.9 to 1.0. After 95 minutes, the charging power dramatically decreases until the battery is fully charged. That is consistent with the charging speeds of batteries observed in the previous study [32, 28]. We use that energy efficiency curve of charging in our simulator. The energy loss of the battery during charging is around 18% of the capacity.

Server calibrations. Figure 5 shows power-performance calibrations by varying the CPU frequency (with DVFS) and workload setting for TPC-C and TPC-H Q5. Note, we have filtered the settings with lower energy efficiency at the same power. The result of server calibration demonstrates the non-energy-proportional power-performance profile, which is consistent with the previous studies [3, 33]. We use these calibrations to further develop a full power-performance profile. We have tried the linear regression and neural network to capture the profile. We find that the neural network is more accurate and thus we use the neural network model to model the power-performance profile. With the neural network model, we are able to solve the problem of finding the optimal throughput for a given power budget. With the same approach, we develop the solution for the dual problem of finding the optimal hardware setting for a given amount of workload.

4.3 Results

Overall comparison. Figure 6 shows the reduction of brown energy consumption by different approaches compared with GO in running TPC-C benchmark and TPC-H Q5. The battery capacity is provisioned to be 20Ah. Overall, all variants of ReinDB have brown energy reduction. Both workload shifting and battery usage demonstrate improvements on reducing the brown energy. Among them, battery usage is much more significant due to the assistance of battery in resolving the mismatch between workload and green supply. Among the tested workloads, the overall brown energy reduction of ReinDB over GO is up to 28% and 57% for TPC-C and TPC-H Q5, respectively. The reduction over green-aware SDE is up to 28% and 53% for TPC-C and TPC-H Q5, respectively.

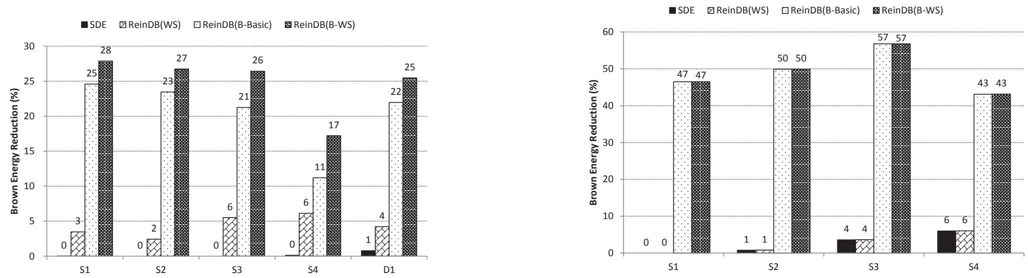
We have made the following observations on the detailed results of TPC-C and TPC-H Q5. First, SDE has only very slight improvement over GO, in contrast with the previous studies [17]. This is because, compared with the previous study, our workload has the relatively tight deadlines and the database server has the limited capacity. Note, previous studies [17, 9] usually assume that data centers have sufficient computational resources for all delayed workloads. Second, due to the non-energy-proportional power-performance profile of the server, workload shifting has reasonable improvement by shifting the workload to the most energy efficient point. The major improvement is obtained when the solar supply increases along the time. Third, battery plays a significant role in storing the green energy for future use. This improvement is obtained in the current setting where the battery capacity is usually less than 1% of the total energy consumption of running the whole trace. Fourth, whether to create sufficiently long idle periods to take advantage of ACPI S3 is important for energy saving. On S1 and S2 of TPC-H Q5, the improvement of SDE and ReinDB(WS) is very small, because GO executes the query in the fastest manner and then creates longer idle periods for ACPI S3. In contrast, battery-aware workload shifting tends to create even longer idle periods than GO.

Figure 7 shows the energy breakdown in the total energy consumption of D1 with TPC-C. We divide the total energy into three categories: *brown* energy consumption and *green* energy consumption by the workload, and battery-related consumption. We further divide the battery-related consumption into two parts: *battery discharging* and *battery loss* denoting the amount of energy discharged from the battery (i.e., used by the workload) and the amount of energy loss during charging the battery. According to the breakdown, integrating the solar energy reduces the brown energy usage by 40% in GO, and ReinDB decreases the brown energy usage by 44-55%. After examining all the experiments, we find that ReinDB with battery sometimes consumes a larger amount of total energy due to battery loss (always less than 1%), but always consumes the least amount of brown energy.

We also investigate the frequency of charging and discharging battery. The time for charging and discharging the battery is usually 14–27% and 18–25% of the total time across different workloads. Potentially, dynamic charging/discharging may hurt the lifetime of batteries. It is our future work to take this issue into consideration for battery-aware workload shifting.

Sensitivity studies. We have further conducted sensitivity studies on the parameters involved in ReinDB design such as battery capacity, solar energy scale, and the prediction accuracy. We present the results of sensitivity studies mainly with D1 (except the experiments on varying slacks with S2).

Figure 8 shows the result for varying the battery capacity. Increasing the battery capacity can further increase the brown energy reduction of ReinDB. However, the improvement



(a) TPC-C (b) TPC-H Q5

Figure 6: The brown energy reduction compared with GO.

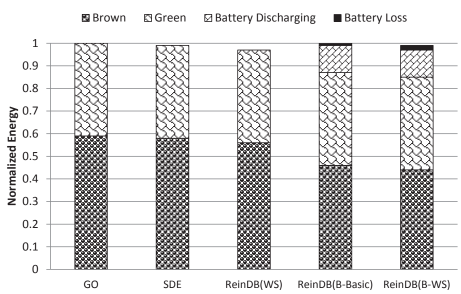


Figure 7: The energy consumption breakdown of TPC-C D1.

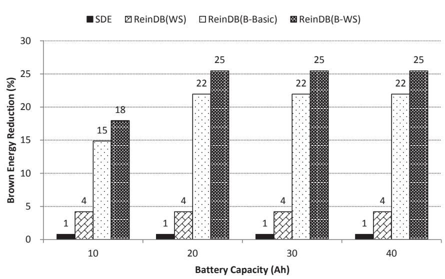


Figure 8: The brown energy reduction compared with GO varying the battery capacity.

is marginally decreased. When the battery capacity is larger than 20Ah, the brown energy reduction becomes stable across different approaches.

Figure 9 shows the result for varying the solar energy scale. The increasing solar energy scale can be achieved by using more effective solar panels and/or by installing solar panels of larger area. As the scale of solar energy increases, the brown energy reduction by ReinDB increases. Particularly, the improvement of workload shifting slightly decreases, and that of battery significant increases.

Figure 10 shows the brown energy reduction by introducing different degrees of prediction errors in the solar supply estimation. 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 ReinDB over other approaches has only a very small degration. We observed similar results by varying the error in workload predictions.

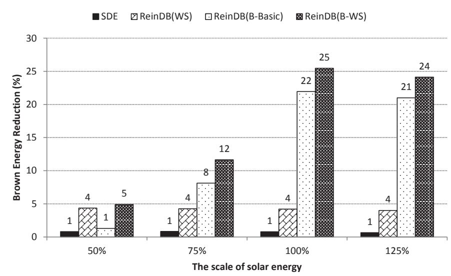


Figure 9: The brown energy reduction compared with GO varying the scale of solar energy.

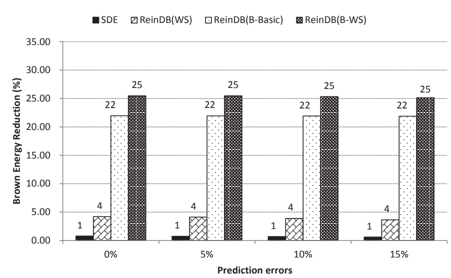


Figure 10: The brown energy reduction compared with GO varying prediction errors.

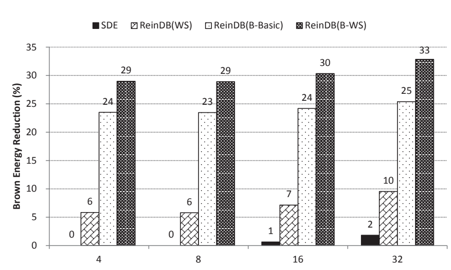


Figure 11: The brown energy reduction compared with GO varying slacks.

Figure 11 shows the results for different slacks for S2. Unlike the survey result, each query has a fixed slack here, and we vary the slack from 4, 8, 16 and 32 seconds. As expected, as the slack increases, the opportunity of utilizing green energy becomes larger and the brown energy reduction increases. It indicates that, the role of users is important in realizing green databases.

5. FUTURE DIRECTIONS

Previous sections have demonstrated the case that a naive integration of renewable energy can severely underutilize the renewable energy; our preliminary work with green-aware optimizations significantly improves the utilization of renewable energy and thus reduces the amount of brown energy usage. ReinDB fosters a set of interesting new research directions. We briefly discuss a few of them.

Making database systems with more green-awareness. The research agenda will go beyond workload scheduling and hardware setting, for example, investigating the green-awareness on query operator implementation, query optimization and execution. Particularly, to realize these opportunities in a data management system, existing system components should be re-visited, and new components should be introduced. We examine the opportunities in two key components of data management systems: query optimizer and query/task scheduler.

When the renewable energy supply is dynamic, i.e., it is not known at the compilation time, a dynamic query optimization phase is required. A vast of dynamic optimization techniques [6] are relevant and should be re-visited under the context of dynamic energy supply.

The query scheduler decides when and where to execute a certain query. In addition to queries, some tasks in the database systems may be possible to defer, including (a) view maintenance, (b) table/index reorganization, and (c) other maintenance tasks. Some queries are not urgent, for example, some routine queries for reporting. On the other hand, speculative executions are also worthwhile on the time slots with extensive renewable energy, especially for recurring queries [13]. We can incrementally compute the partial results at the low brown energy usage for future reuse.

Beyond re-design of database system components, there are challenges in other research and engineering fields, for example, manufacturing design on integrating solar panels and other components, and economic analysis on solar panels and battery provision.

Automaticity in exploiting green-awareness. Introducing green-awareness into data management systems has further added a series of tuning knobs, which might increase the total ownership cost. The system (re-)design should take automaticity into consideration. The current design of ReinDB relies on the workload and green supply prediction. It might be desirable to develop feedback control for the prediction error. Other issues include real-time monitoring on brown and energy supply, and automatic calibrations on accurate battery models. We believe that careful system design and modeling can alleviate the complexity of introducing green-awareness into the system.

Green Energy Powered Data Centers. This study considers individual scale-out servers in the data center. When we consider the scale of powering data centers with green energy, many techniques/optimizations that have been developed for a single database server are still relevant. The key challenge is to adapt them into a much larger scale. Beyond those green-aware optimizations, we also have other issues to consider. For example, we can consider other ways of storing the renewable energy, e.g., water pumps and thermal. We need to consider various factors like geographical features, economics, and charging/discharging efficiency for the suitable energy storage. One may leverage batteries within the data center [27].

When Database Meets Power Grid. The penetration of renewable energy in power grids is a recent and hot issue. Although the penetration in most power grids is low, we expect the penetration will go up in the future. That means, the electricity from public grids is mixed of brown and green energy. Moreover, the ratio between the two supply sources varies in power grids, depending on where users draw the electricity. That information is necessary for green-aware computing. Smart grid may be able to deliver this information to users (e.g., the LoCal project [16]). With the implementation of smart grid, the information on green energy can be delivered to users so that users can adjust their computing behavior according to green supply.

6. RELATED WORK

This section reviews the most relevant related work on renewable energy aware computing and energy efficient data management.

Renewable energy aware computing. Renewable energy aware computing has become a hot research issue in data centers. Most of the previous studies focus on integrating renewable energy into data centers, and adopt supply driven execution by considering the job slacks. Íñigo Goiri et al. leveraged renewable energy to handle scientific workloads [9]. Baris Aksanli et al. developed green-aware scheduling for both online services and batch jobs in a single data center [1]. Íñigo Goiri et al. integrated green awareness into Hadoop (namely GreenHadoop [10]). Krioukov et al. [17] advocated a supply-following computing paradigm for data intensive applications. Chen et al. [4] proposed to schedule and migrate workloads across multiple geographically distributed data centers in order to fully utilize the renewable energy in different regions. Most studies [9, 17] assume that the aggregated workload does not exceed the data center capacity, and defer the task execution as much as possible. In contrast, this paper implements SDE on a server by considering the server processing capacity and energy efficient hardware settings. Moreover, we improve SDE by considering the efficiency of brown and green energy usage. While our work targets at a server, our optimizations are applicable to individual scale-out servers within data centers. Some attention has been paid to leverage battery to store renewable energy [11, 2]. Their battery usage is the same as our baseline algorithm, without considering the energy efficiency at different power levels.

Energy saving database techniques. Previous studies [3, 33] have shown that database systems are far from energy-proportional. Hardware and software approaches have been proposed to improve the energy efficiency. Low cost and low power hardware have demonstrated high energy efficiency for some workloads [22, 29]. Dynamic adjustment for hardware settings [19, 20] is useful to improve the energy efficiency of databases. Lang et al. [18] explored various key parameters in building energy-efficient database clusters. While those techniques are green oblivious, they are relevant to our SDE scheme on finding the most energy efficient point of a database server. We refer readers to a recent survey [34] on other energy efficient data management techniques. To the best of our knowledge, this is the first work of attempting to integrate green energy into a database server.

7. CONCLUSION REMARKS

This paper proposes ReinDB that integrates renewable energy supply into database systems in order to minimize the brown energy usage and environmental pollution. We have developed green supply driven execution on the database server, and further developed energy efficient optimizations for both green and brown energy usage. Our preliminary results demonstrate that ReinDB significantly reduces the brown energy usage (up to 57% and 53% reduction compared with green-oblivious and basic green-aware approaches, respectively). We consider ReinDB as an important step of reducing the carbon footprint of database systems, which is complementary to the current main-stream research on energy efficient databases. While this paper focuses on the design and implementation of a green database server, we expect to see new green data management systems with renewable energy at larger scales (such as clusters and data centers) emerging in the future.

Acknowledgement

The authors would like to thank anonymous reviewers for their valuable comments, and Dr. Lee Sau-Lai for designing the survey presented in Section 2. The work of Cheng Chen was done when he was a visiting student in Nanyang Technological University. 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".

8. REFERENCES

[1] B. Aksanli, J. Venkatesh, L. Zhang, and T. Rosing. Utilizing green energy prediction to schedule mixed batch and service jobs in data centers. In *Proceedings of the 4th Workshop on Power-Aware Computing and Systems*, HotPower '11, pages 5:1–5:5, New York, NY, USA, 2011. ACM.

[2] A. Bar-Noy, M. P. Johnson, and O. Liu. Approximation and online algorithms. chapter Peak Shaving through Resource Buffering. 2009.

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

[4] C. Chen, B. He, and X. Tang. Green-aware workload scheduling in geographically distributed data centers. In *Proceedings of the 2012 IEEE 4th International Conference on Cloud Computing Technology and Science*, CloudCom '12, pages 1–8, Taipei, Taiwan, 2012. IEEE Computer Society.

[5] Database Test Suite. http://osdldbt.sourceforge.net/, 2012.

[6] A. Deshpande, Z. Ives, and V. Raman. Adaptive query processing. *Found. Trends databases*, 1(1):1–140, 2007.

[7] S. S. Elnaffar. *Towards workload-aware dbmss: identifying workload type and predicting its change*. PhD thesis, Kingston, Ont., Canada, Canada, 2004.

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

[9] I. n. Goiri, R. Beauchea, K. Le, T. D. Nguyen, M. E. Haque, J. Guitart, J. Torres, and R. Bianchini. Greenslot: scheduling energy consumption in green datacenters. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 20:1–20:11, New York, NY, USA, 2011. ACM.

[10] I. n. Goiri, K. Le, T. D. Nguyen, J. Guitart, J. Torres, and R. Bianchini. Greenhadoop: leveraging green energy in data-processing frameworks. In *Proceedings of the 7th ACM european conference on Computer Systems*, EuroSys '12, pages 57–70, New York, NY, USA, 2012. ACM.

[11] S. Govindan, A. Sivasubramaniam, and B. Urgaonkar. Benefits and limitations of tapping into stored energy for datacenters. In *Computer Architecture (ISCA), 2011 38th Annual International Symposium on Computer Architecture*, pages 341 –351, june 2011.

[12] S. Harizopoulos, M. A. Shah, J. Meza, and P. Ranganathan. Energy efficiency: The new holy grail of data management systems research. In *Proc. of CIDR'09*, pages 1–8, 2009.

[13] 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 *Proceedings of the 1st ACM symposium on Cloud computing*, SoCC '10, pages 63–74, New York, NY, USA, 2010. ACM.

[14] S. Jebaraj and S. Iniyan. A review of energy models. *Renewable and Sustainable Energy Reviews*, 10(4):281 – 311, 2006.

[15] A. Kansal and F. Zhao. Fine-grained energy profiling for power-aware application design. *SIGMETRICS Perform. Eval. Rev.*, 36(2):26–31, Aug. 2008.

[16] R. H. Katz, D. E. Culler, S. Sanders, S. Alspaugh, Y. Chen, S. Dawson-Haggerty, P. Dutta, M. He, X. Jiang, L. Keys, A. Krioukov, K. Lutz, J. Ortiz, P. Mohan, E. Reutzel, J. Taneja, J. Hsu, and S. Shankar. An information-centric energy infrastructure: The berkeley view. *Sustainable Computing: Informatics and Systems*, 1(1):7–22, 2011.

[17] A. Krioukov, C. Goebel, S. Alspaugh, Y. Chen, D. Culler, and Y. Katz. Integrating renewable energy using data analytics systems: Challenges and opportunities. In *Bulletin of the IEEE Computer Society Technical Committee*, pages 1–9, 2011.

[18] W. Lang, S. Harizopoulos, J. M. Patel, M. A. Shah, and D. Tsirogiannis. Towards energy-efficient database cluster design. *PVLDB*, 5(11):1684–1695, 2012.

[19] W. Lang, R. Kandhan, and J. M. Patel. Rethinking query processing for energy efficiency: Slowing down towin the race. *In Bulletin of the IEEE Computer Society Technical Committee*, pages 1–12, 2011.

[20] W. Lang and J. M. Patel. Towards ecofriendly database management systems. In *Proc. of CIDR'09*, pages 1–8, 2009.

[21] W. Lang and J. M. Patel. Energy management for mapreduce clusters. *Proc. VLDB Endow.*, 3(1-2):129–139, Sept. 2010.

[22] W. Lang, J. M. Patel, and S. Shankar. Wimpy node clusters: what about non-wimpy workloads? In *Proceedings of the Sixth International Workshop on Data Management on New Hardware*, DaMoN '10, pages 47–55, New York, NY, USA, 2010. ACM.

[23] C. Li, W. Zhang, C.-B. Cho, and T. Li. Solarcore: Solar energy driven multi-core architecture power management. In *Proceedings of the 2011 IEEE 17th International Symposium on High Performance Computer Architecture*, HPCA '11, pages 205–216, Washington, DC, USA, 2011. IEEE Computer Society.

[24] D. Linden and T. B. Reddy. *Handbook of Batteries*. McGraw-Hill, 2001.

[25] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew. Greening geographical load balancing. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, SIGMETRICS '11, pages 233–244, New York, NY, USA, 2011. ACM.

[26] D. Meisner, C. Sadler, L. Barroso, W. Weber, and T. Wenisch. Power management of online data-intensive services. In *Computer Architecture (ISCA), 2011 38th*

Annual International Symposium on Computer Architecture, pages 319 –330, june 2011.

[27] D. S. Palasamudram, R. K. Sitaraman, B. Urgaonkar, and R. Urgaonkar. Using batteries to reduce the power costs of internet-scale distributed networks. In *Proceedings of the Third ACM Symposium on Cloud Computing*, SoCC '12, pages 11:1–11:14, New York, NY, USA, 2012. ACM.

[28] P. Rong and M. Pedram. An analytical model for predicting the remaining battery capacity of lithium-ion batteries. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 14(5):441 –451, may 2006.

[29] D. Schall and V. Hudlet. Wattdb: an energy-proportional cluster of wimpy nodes. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, SIGMOD '11, pages 1229–1232, New York, NY, USA, 2011. ACM.

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

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

[32] R. S. Treptow. The lead-acid battery: Its voltage in theory and in practice. *Journal of Chemical Education*, 79(3):334, 2002.

[33] D. Tsirogiannis, S. Harizopoulos, and M. A. Shah. Analyzing the energy efficiency of a database server. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pages 231–242, New York, NY, USA, 2010. ACM.

[34] J. Wang, L. Feng, W. Xue, and Z. Song. A survey on energy-efficient data management. *SIGMOD Rec.*, 40(2):17–23, Sept. 2011.

[35] D. Wu, B. He, X. Tang, J. Xu, and M. Guo. Ramzzz: rank-aware dram power management with dynamic migrations and demotions. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '12, pages 32:1–32:11, Los Alamitos, CA, USA, 2012. IEEE Computer Society Press.

[36] Y. Zhang, Y. Wang, and X. Wang. Greenware: Greening cloud-scale data centers to maximize the use of renewable energy. In F. Kon and A.-M. Kermarrec, editors, *Middleware 2011*, volume 7049 of *Lecture Notes in Computer Science*, pages 143–164. Springer Berlin Heidelberg, 2011.

APPENDIX

A. DEMO PLAN

The goal of our demo is to demonstrate the design of our database system with the integration of solar energy, and the effectiveness of our green-aware optimizations.

We plan to conduct the demonstration in two ways: real system prototype and simulation.

We are developing the proof-of-concept system prototype, by assembling the components from the commodity markets. Their configurations and prices are listed in Table 1. The extra hardware cost for supporting the solar energy supply is around 45% of the server budget. It is within the increment tolerance observed in the survey (on average 64.1%).

Table 1: Hardware components in our proof-of-concept system.

Hardware	Description	Price (USD)
Computer	HP Z420 Workstation	2000
Charge controller	Provista Charge controller ISC3030	103
Battery	Rocket 65AH Battery	93
Power Inverter	Power Inverter (SUN-1200W)	74
Power meter	Watts Up Pro	156
Solar panel	140W Solar Panel polycrystalline	479

At the time of writing the paper, we have not got a switch from our vendor. Thus, we cannot dynamically switch between brown energy supply from public grid, green energy supply and battery. We hope to get the switch from the vendor soon. If the switch is still not available during the conference, we will demonstrate ReinDB by simulating brown energy supply with the battery supply. We first fully charge the battery and then run ReinDB at no-battery mode. When the solar power is smaller than the demand, ReinDB draws power from the battery.

In the real system prototype demonstration, we show how the solar powered system works in practice, including the system components and their configuration as well as the runtime status of the system. The system prototype is deployed in our campus in Singapore (as illustrated in Figure 12). The solar panel is installed on the roof of a building, and the energy is redirected to our office downstairs. The figure is for illustration purposes and the workstation is actually placed in our office, with the battery supply and the green energy supply generated from the solar panel on the roof. Since it is not feasible to bring the system prototype to the conference venue, the demonstration will be conducted with a short video.

Figure 12: The system prototype: 1. solar panel, 2. workstation, 3. charge controller, 4. battery, 5. power meter and 6.inverter.

In the simulation, we demonstrate our green database system under different user profiles on slacks, workloads, hardware configuration and solar traces. Specifically, we will compare the performance and energy consumption of our green database with other baseline approaches. Visualization will be developed to show the runtime statistics including users (profiles), workloads (transactions and their deadlines), performance (throughput and response time), power, energy consumption (total energy consumption, brown/green energy consumption) and solar power etc. Our simulation demo allows users to specify their profiles and solar traces.