Renewable and Cooling Aware Workload Management for Sustainable Data Centers*

Zhenhua Liu[$], Yuan Chen[†], Cullen Bash[†], Adam Wierman[$],
Daniel Gmach[†], Zhikui Wang[†], Manish Marwah[†], Chris Hyser[†]
[$]California Institute of Technology, Pasadena, CA, USA, E-mail: {zhenhua, adamw}@caltech.edu
[†]HP Labs, Palo Alto, CA, USA, E-mail: firstname.lastname@hp.com

ABSTRACT

Recently, the demand for data center computing has surged, increasing the total energy footprint of data centers worldwide. Data centers typically comprise three subsystems: IT equipment provides services to customers; power infrastructure supports the IT and cooling equipment; and the cooling infrastructure removes heat generated by these subsystems. This work presents a novel approach to model the energy flows in a data center and optimize its operation. Traditionally, supply-side constraints such as energy or cooling availability were treated independently from IT workload management. This work reduces electricity cost and environmental impact using a holistic approach that integrates renewable supply, dynamic pricing, and cooling supply including chiller and outside air cooling, with IT workload planning to improve the overall sustainability of data center operations. Specifically, we first predict renewable energy as well as IT demand. Then we use these predictions to generate an IT workload management plan that schedules IT workload and allocates IT resources within a data center according to time varying power supply and cooling efficiency. We have implemented and evaluated our approach using traces from real data centers and production systems. The results demonstrate that our approach can reduce both the recurring power costs and the use of non-renewable energy by as much as 60% compared to existing techniques, while still meeting the Service Level Agreements.

Categories and Subject Descriptors

C.0 [**Computer Systems Organization**]: General

Keywords

sustainable data center, renewable energy, demand shaping, scheduling, cooling optimization

1. INTRODUCTION

Data centers are emerging as the "factories" of this generation. A single data center requires a considerable amount of electricity and data centers are proliferating worldwide as a result of increased demand for IT applications and services. As a result, concerns about the growth in energy usage and emissions have led to social interest in curbing their energy consumption. These concerns have led to research efforts in both industry and academia. Emerging solutions include the incorporation of renewable on-site energy supplies as in Apple's new North Carolina data center, and alternative cooling supplies as in Yahoo's New York data center. The problem addressed by this paper is how to use these resources most effectively during the operation of data centers.

Most of the efforts toward this goal focus on improving the efficiency in one of the three major data center silos: (i) IT, (ii) cooling, and (iii) power. Significant progress has been made in optimizing the energy efficiency of each of the three silos enabling sizeable reductions in data center energy usage, e.g., [1, 2, 3, 4, 5, 6, 7, 8]; however, the integration of these silos is an important next step. To this end, a second generation of solutions has begun to emerge. This work focuses on the integration of different silos [9, 10, 11, 12]. An example is the dynamic thermal management of air-conditioners based on load at the IT rack level [9, 13]. However, to this point, supply-side constraints such as renewable energy and cooling availability are largely treated independently from workload management such like scheduling. Particularly, current workload management are not designed to take advantage of time variations in renewable energy availability and cooling efficiencies. The work in [14] integrates power capping and consolidation with renewable, but they do not shift workloads to align power demand with renewable supply.

The potential of integrated, dynamic approaches has been realized in some other domains, e.g., cooling management solutions for buildings that predict weather and power prices to dynamically adapt the cooling control have been proposed [15]. The goal of this paper is to start to realize this potential in data centers. Particularly, the potential of an integrated approach can be seen from the following three observations:

First, most data centers support a range of IT workloads, including both critical interactive applications that run 24x7 such like Internet services, and delay tolerant, batch-style applications as scientific applications, financial analysis, and image processing, which we refer to as batch workloads or batch jobs. Generally, batch workloads can be scheduled to run anytime as long as they finish before deadlines. This enables significant flexibility for workload management.

Second, the availability and cost of power supply, e.g., renewable energy supply and electricity price, is often dynamic over time, and so dynamic control of the supply mix can help reduce CO_2 emissions and offset costs. Thus, thoughtful workload management can have a great impact on energy usage and costs by scheduling batch workloads in a manner that follows the renewable availability.

Third, many data centers nowadays are cooled by multiple means through a cooling micro grid combining traditional mechanical chillers, airside economizers, and waterside econ-

*This work is done during Zhenhua Liu's internship at HP Labs. Zhenhua Liu and Adam Wierman are partly supported by NSF grant CNS 0846025 and DoE grant DE-EE0002890.

SIGMETRICS'12, June 11–15, 2012, London, England, UK.

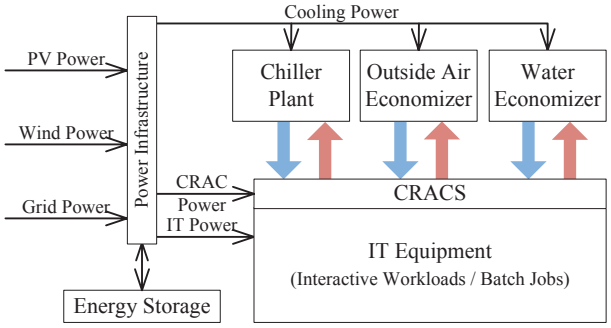


Figure 1: Sustainable Data Center

omizers. Within a micro grid, each cooling approach has a different efficiency and capacity that depends on IT workload, cooling generation mechanism and external conditions including outside air temperature and humidity, and may vary with the time of day. This provides opportunities to optimize cooling cost by "shaping" IT demand according to time varying cooling efficiency and capacity.

The three observations above highlight that there is considerable potential for integrated management of the IT, cooling, and power subsystems of data centers. Providing such an integrated solution is the goal of this work. Specifically, we provide a novel workload scheduling and capacity management approach that integrates energy supply (renewable energy supply, dynamic energy pricing) and cooling supply (chiller cooling, outside air cooling) into IT workload management to improve the overall energy efficiency and reduce the carbon footprint of data center operations.

A key component of our approach is demand shifting, which schedules batch workloads and allocates IT resources within a data center according to the availability of renewable energy supply and the efficiency of cooling. This is a complex optimization problem due to the dynamism in the supply and demand and the interaction between them. To see this, given the lower electricity price and temperature of outside air at night, batch jobs should be scheduled to run at night; however, because more renewable energy like solar is available around noon, we should do more work during the day to reduce electricity bill and environmental impact.

At the core of our design is a model of the costs within the data center, which is used to formulate a constrained convex optimization problem. The workload planner solves this optimization to determine the optimal demand shifting. The optimization-based workload management has been popular in the research community recently, e.g., [16, 17, 18, 19, 3, 11, 20, 21]. The key contributions of the formulation considered here compared to the prior literature are (i) the addition of a detailed cost model and optimization of the cooling component of the data center, which is typically ignored in previous designs; (ii) the consideration of both interactive and batch workloads; and (iii) the derivation of important structural properties of the optimal solutions to the optimization.

In order to validate our integrated design, we have implemented a prototype of our approach for a data center that includes solar power and outside air cooling. Using our implementation, we perform a number of experiments on a real testbed to highlight the practicality of the approach (Section 5). In addition to validating our design, our experiments are centered on providing insights into the following questions:

(1) How much benefit (reducing electricity bill and environmental impact) can be obtained from our renewable and cooling-aware workload management planning?

(2) Is net-zero[1] grid power consumption achievable?

[1]By "net-zero" we mean that the total energy usage over a fixed period is less than or equal to the local total renewable generation during that period. Note that this does not mean that no power from the grid is used during this period.

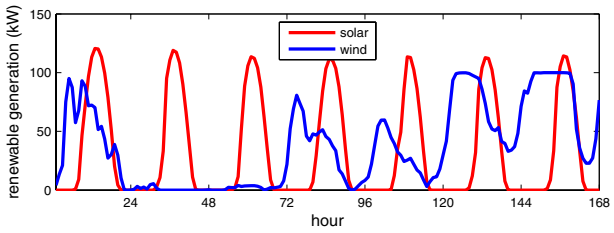


Figure 2: One week renewable generation

(3) Which renewable source is more valuable? What is the optimal renewable portfolio?

2. SUSTAINABLE DATA CENTER OVERVIEW

Figure 1 depicts an architectural overview of a sustainable data center. The *IT equipment* includes servers, storage and networking switches that support applications and services hosted in the data center. The *power infrastructure* generates and delivers power for the IT equipment and cooling facility through a power micro grid that integrates grid power, local renewable generation such as photovoltaic (PV) and wind, and energy storage. The *cooling infrastructure* provides, delivers and distributes the cooling resources to extract the heat from the IT equipment. In this example, the cooling capacity is delivered to the data center through the Computer Room Air Conditioning (CRAC) Units from the cooling micro grid that consists of air economizer, water economizer and traditional chiller plant. We discuss these three key subsystems in detail in the following sections.

2.1 Power Infrastructure

Although renewable energy is in general more sustainable than grid power, the supply is often time varying in a manner that depends on the source of power, location of power generators, and the local weather conditions. Figure 2 shows the power generated from a 130kW PV installation for an HP data center and a nearby 100kW wind turbine in California, respectively. The PV generation shows regular variation while that from the wind is much less predictable. How to manage these supplies is a big challenge for application of renewable energy in a sustainable data center.

Despite the usage of renewable energy, data centers must still rely on non-renewable energy, including grid power and on-site energy storage, due to availability concerns. Grid power can be purchased at either a pre-defined fixed rate or an on-demand time-varying rate, and Figure 3 shows an example of time-varying electricity price over 24 hours. There might be an additional charge for the peak demand.

Local energy storage technologies can be used to store and smooth out the supply of power for a data center. A variety of technologies are available [22], including flywheels, batteries, and other systems. Each has its costs, advantages and disadvantages. Energy storage is still quite expensive and there is power loss associated with energy conversion and charge/discharge. Hence, it is critical to maximize the use of the renewable energy that is generated on site. An ideal scenario is to maximize the use of renewable energy while minimizing the use of storage.

2.2 Cooling Supply

Due to the ever-increasing power density of IT equipment in today's data centers, a tremendous amount of electricity is used by the cooling infrastructure. According to [23], a

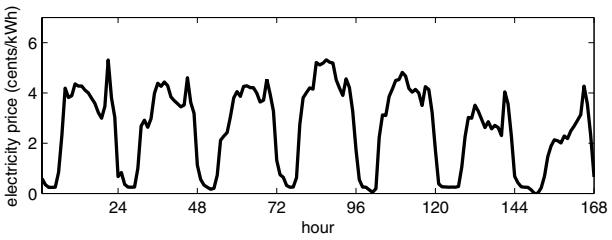


Figure 3: One week real-time electricity price

significant amount of data center power goes to the cooling system (up to 1/3) including CRAC units, pumps, chiller plant and cooling towers.

Lots of work has been done to improve the cooling efficiency through, e.g., smart facility design, real-time control and optimization [8, 7]. Traditional data centers use chillers to cool down the returned hot water from CRACs via mechanical refrigeration cycles since they can provide high cooling capacity continuously. However, compressors within the chillers consume a large amount of power [24, 25]. Recently, "chiller-less" cooling technologies have been adopted to remove or reduce the dependency on mechanical chillers. In the case with water-side economizers, the returned hot water is cooled down by components such as dry coolers or evaporative cooling towers. The cooling capacity may also be generated from cold water from seas or lakes. In the case of air economizers, cold outside air may be introduced after filtering and/or humidification/de-humidification to cool down the IT equipment directly while hot air is rejected into the environment.

However, these so-called "free" cooling approaches are actually not free [24]. First, there is still a non-negligible energy cost associated with these approaches, e.g., blowers driving outside air through data center need to work against air flow resistance and therefore consume power. Second, the efficiency of these approaches is greatly affected by environmental conditions such as ambient air temperature and humidity, compared with that of traditional approaches based on mechanical chillers. The cooling efficiency and capacity of the economizers can vary widely along with time of the day, season of the year, and geographical locations of the data centers. These approaches are usually complemented by more stable cooling resources such as chillers, which provides opportunities to optimize the cooling power usage by "shaping" IT demand according to cooling efficiencies.

2.3 IT Workload

There are many different workloads in a data center. Most of them fit into two classes: interactive, and non-interactive or batch. The interactive workloads such as Internet services or business transactional applications typically run 24x7 and process user requests, which have to be completed within a certain time (response time), usually within a second. Non-interactive batch jobs such as scientific applications, financial analysis, and image processing are often delay tolerant and can be scheduled to run anytime as long as progress can be made and the jobs finish before the deadline (completion time). This deadline is much more flexible (several hours to multiple days) than that of interactive workload. This provides great optimization opportunities for workload management to "shape" non-interactive batch workloads based on the varying renewable energy and cooling supply.

Interactive workloads are characterized by stochastic properties for request arrival, service demand, and Service Level Agreements (SLAs, e.g., thresholds of average response time or percentile delay). Figure 4 shows a 7-day normalized CPU usage trace for a popular photo sharing and storage web service, which has more than 85 million registered users in 22 countries. We can see that the workload shows significant

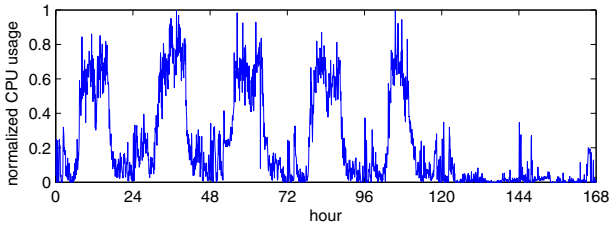


Figure 4: One week interactive workload

variability and exhibits a clear diurnal pattern, which is typical for data center interactive workloads.

Batch jobs are defined in terms of total resource demand (e.g, CPU hours), starting time, completion time as well as maximum resource consumption (e.g., a single thread program can use up to 1 CPU). Conceptually, a batch job can run at anytime on many different servers as long as it finishes before the specified completion time. Our integrated management approach exploits this flexibility to make use of renewable energy and efficient cooling when available.

3. MODELING AND OPTIMIZATION

As discussed above, time variations in renewable energy supply availability and cooling efficiencies provide both opportunities and challenges for managing IT workloads in data centers. In this section, we present a novel design for renewable and cooling aware workload management that exploits opportunities available to improve the sustainability of data centers. In particular, we formulate an optimization problem for adapting the workload scheduling and capacity allocation to varying supply from power and cooling infrastructure.

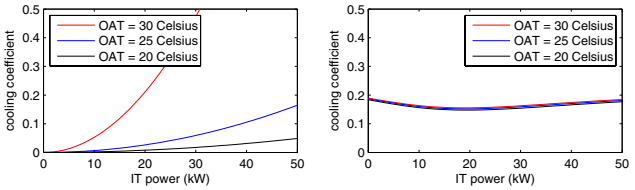
3.1 Optimizing the cooling substructure

We first derive the optimal cooling substructure when multiple cooling approaches are available in the substructure. We consider two cooling approaches: the outside air cooling which supplies most of the cooling capacity, and cooling through mechanical chillers which guarantees availability of cooling capacity. By exploring the heterogeneity of the efficiency and cost of the two approaches, we represent the minimum cooling power of the substructure as a function of the IT heat load.

In the following discussion, we define *cooling coefficient* as the cooling power divided by the IT power to represent the cooling efficiency. By cooling capacity we mean how much heat the cooling system can extract from the IT equipment and reject into the environment. In the case of outside air cooling, the cold air from outside is assumed pushed into the return ends of the CRAC units while the hot air from the outlets of the server racks is exhausted to the ambient environment through ducts.

Outside Air Cooling

The energy usage of outside air cooling is mainly the power consumed by blowers, which can be approximated as a cubic function of the blower speed [26, 24]. We assume that capacity of the outside air cooling is under tight control, e.g., through blower speed tuning, to avoid over-provisioning. Then the outside air capacity is equal to the IT heat load at the steady state when the latter does not exceed the total air cooling capacity. Based on basic heat transfer theory [27], the cooling capacity is proportional to the air volume flow rate. The air volume flow rate is approximately proportional to blower speed according to the general fan laws [26, 24]. Therefore, outside air cooling power can be defined as a function of IT power d as $f_a(d) = kd^3, 0 \leq d \leq \overline{d}, k > 0$, which is a convex function. The parameter k depends on the temperature difference, $(t_{RA} - t_{OA})$, based again on heat transfer



(a) Outside air cooling (b) Chiller cooling

Figure 5: Cooling coefficient comparison, for conversion, 20°C=68°F, 25°C=77°F, 30°C=86°F

theory, where t_{OA} is the outside air temperature (OAT) and t_{RA} is the temperature of the (hot) exhausting air from the IT racks. The maximum capacity of this cooling system can be modeled as $\overline{d} = C(t_{RA} - t_{OA})$. The parameter $C > 0$ is the maximum capacity of the air, which is proportional to the maximal outside air mass flow rate when the blowers run at the highest speed. As one example, Figure 5(a) shows the cooling coefficient for an outside air cooling system (assuming the exhausting air temperature is 35°C/95°F) under different outside air temperatures.

Chilled Water Cooling

First-principle models of chilled water cooling systems, including the chiller plant, cooling towers, pumps and heat exchangers, are complicated [27, 13, 25]. In this paper, we consider an empirical chiller efficiency model that was built on actual measurement of an operational chiller [25]. Figure 5(b) shows the cooling coefficient of the chiller. Different from the outside air cooling, the chiller cooling coefficient does not change much with OAT and the variation over different IT load is much smaller than that under outside air cooling. In the following analysis, the chiller power consumption is approximated as $f_c(d) = \gamma d$, where d is again the IT power and $\gamma > 0$ is a constant depending on the chiller. Our analysis also applies to the case of any arbitrary strictly increasing and convex chiller cooling function [28].

Cooling optimization

As shown in Figure 5, the efficiency of outside air cooling is more sensitive to IT load and the OAT than is that of chiller cooling. Furthermore, the cost of outside air cooling is higher than that of the chiller when the IT load exceeds a certain value because its power increases very fast (superlinearly) as the IT power increases, in particular for high ambient temperatures. The heterogeneous cooling efficiencies of the two approaches and the varying properties along with air temperature and heat load provide opportunities to optimize the cooling cost by using proper cooling capacity from each cooling supply as we discuss below, or by manipulating the heat load through demand shaping as we show in later sections.

For a given IT power d and outside air temperature t_{OA}, there exists an optimal cooling capacity allocation between outside air cooling and chiller cooling. Assume the cooling capacities provided by the chiller and outside air are d_1 and d_2 respectively ($d_1 = d - d_2$). From the cooling models introduced above, the optimal cooling power consumption is

$$c(d) = \min_{d_2 \in [0, \overline{d}]} \gamma(d - d_2)^+ + kd_2^3 \qquad (1)$$

This can be solved analytically, which yields

$$d_2^* = \begin{cases} d & \text{if } d \le d_s \\ d_s & \text{otherwise} \end{cases}$$

where $d_s = \min\left\{\sqrt{\gamma/3k}, \overline{d}\right\}$, and the optimal outside air cooling capacity is $d_1^* = d - d_2^*$. So,

$$c(d) = \begin{cases} kd^3 & \text{if } d \le d_s \\ kd_s^3 + \gamma(d - d_s) & \text{otherwise} \end{cases} \qquad (2)$$

is the cooling power of the optimal substructure, which is used for the optimization in later sections. Figure 6 illustrates the relationship between cooling power and IT load for different ambient temperatures. We see that the cooling power is a convex function of IT power, higher with hotter outside air. We also prove that the optimal $c(d)$ is a convex function of d in [28], which is important for using this as a component of the overall optimization for workload management.

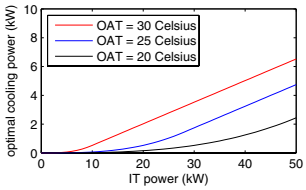


Figure 6: Optimal cooling power

3.2 System Model

We consider a discrete-time model whose timeslot matches the timescale at which the capacity provisioning and scheduling decisions can be updated. There is a (possibly long) time period we are interested in, $\{1, 2, ..., T\}$. In practice, T could be a day and a timeslot length could be 1 hour. The management period can be either static, e.g., to perform the scheduling every day for the execution of the next day, or dynamic, e.g., to create a new plan if the old scheduling differs too much from the actual supply and demand. The goal of the workload management is at each time t to:

(i) Make the scheduling decision for each batch job;
(ii) Choose the energy storage usage;
(iii) Optimize the cooling infrastructure.

We assume the renewable supply at time t is $r(t)$, which may be a mix of different renewables, such as wind and PV solar. We denote the grid power price at time t by $p(t)$ and assume $p(t) > 0$ without loss of generality. If at some time t we have negative price, we will use up the total capacity at this timeslot, then we only need to make capacity decisions for other timeslots, see [28] for more details. To model energy storage, we denote the energy storage level at time t by $es(t)$ with initial value $es(0)$ and the discharge/charge at time t by $e(t)$, where positive or negative values mean discharge or charge, respectively. Also, there is a loss rate $\rho \in [0, 1]$ for energy storage. We therefore have the relation $es(t+1) = \rho(es(t) - e(t))$ between successive timeslots and we require $0 \le es(t) \le ES, \forall t$, where ES is the energy storage capacity. Though there are more complex energy storage models [29], they are beyond the scope of the paper.

Assume that there are I interactive workloads. For interactive workload i, the arrival rate at time t is $\lambda_i(t)$, the mean service rate is μ_i and the target performance metrics (e.g., average delay, or 95th percentile delay) is rt_i. In order to satisfy these targets, we need to allocate interactive workload i with IT capacity $a_i(t)$ at time t. Here $a_i(t)$ is derived from analytic models (e.g., M/GI/1/PS, M/M/k) or system measurements as a function of $\lambda_i(t)$ because performance metrics generally improve as the capacity allocated to the workload increases, hence there is a sharp threshold for $a_i(t)$. Note that our solution is quite general and does not depend on a particular model.

Assume there are J classes of batch jobs. Class j batch jobs have total demand B_j, maximum parallelization MP_j, starting time S_j and deadline E_j. Let $b_j(t)$ denote the amount of capacity allocated to class j jobs at time t. We

have $0 \leq b_j(t) \leq MP_j, \forall t$ and $\sum_t b_j(t) \leq B_j$. Given the above definitions, the total IT demand at time t is given by

$$d(t) = \Sigma_i a_i(t) + \Sigma_j b_j(t). \quad (3)$$

When taking into consideration the server power model, we can further transform $d(t)$ into power demand, as in Section 4.1. We assume the total IT capacity is D, so $0 \leq d(t) \leq D, \forall t$. Note here $d(t)$ is not constant, but instead time-varying as a result of dynamic capacity provisioning.

3.3 Cost and Revenue Model

The cost of a data center includes both capital and operating costs. Our model focuses on the operational electricity cost. Meanwhile, by servicing the batch jobs, the data center can obtain revenue. We model the data center cost by combining the energy cost and revenue from batch jobs. Note that, to simplify the model, we do not include the switching costs associated with cycling servers in and out of power-saving modes; however, the approach of [3] provides a natural way to incorporate such costs if desired.

To capture the variation of the energy cost over time, we let $g(t, d(t), e(t))$ denote the energy cost of the data center at time t given the IT power $d(t)$, optimal cooling power, renewable generation, electricity price, and energy storage usage $e(t)$. For any t, we assume that $g(t, d(t), e(t))$ is non-decreasing in $d(t)$, non-increasing in $e(t)$, and jointly convex in $d(t)$ and $e(t)$.

This formulation is quite general, and captures, for example, the common charging plan of a fixed price per kWh plus an additional "demand charge" for the peak of the average power used over a sliding 15 minute window [30], in which case the energy cost function consists of two parts:

$$p(t)\left(d(t) + c(d(t)) - r(t) - e(t)\right)^+$$

and

$$p_{peak}\left(\max_t \left(d(t) + c(d(t)) - r(t) - e(t)\right)^+\right),$$

where $p(t)$ is the fixed/variable electricity price per kWh, and p_{peak} is the peak demand charging rate. We could also include a sell-back mechanism and other charging policies. Additionally, this formulation can capture a wide range of models for server power consumption, e.g., energy costs as an affine function of the load, see [1], or as a polynomial function of the speed, see [4, 31].

We model only the variable component of the revenue[2], which comes from the batch jobs that are chosen to be run. Specifically, the data center gets revenue $\mathcal{R}(\mathbf{b})$, where $\mathbf{b}$ is the matrix consisting of $b_j(t), \forall j, \forall t$. In this paper, we focus on the following, simple revenue function

$$\mathcal{R}(\mathbf{b}) = \Sigma_j R_j \left(\Sigma_{t\in[S_j,E_j]} b_j(t)\right),$$

where R_j is the per-job revenue. $\left(\Sigma_{t\in[S_j,E_j]} b_j(t)\right)$ captures the amount of class j jobs finished before their deadlines.

3.4 Optimization Problem

We are now ready to formulate the renewable and cooling aware workload management optimization problem. Our optimization problem takes as input the renewable supply $r(t)$, electricity price $p(t)$, optimal cooling substructure $c(d(t))$, and IT workload demand $a_i(t), B_j$ and related information (the starting time S_j, deadline E_j, maximum parallization MP_j), IT capacity D, energy storage capacity ES and loss rate ρ, and generates an optimal schedule of each timeslot for batch jobs $b_j(t)$ and energy storage usage $e(t)$, according to the availability of renewable power and cooling supply such that specified SLAs (e.g., deadlines) and operational goals (e.g., minimizing operational costs) are satisfied.

This is captured by the following optimization problem:

$$\min_{\mathbf{b},\mathbf{e}} \Sigma_t g(t, d(t), e(t)) - \Sigma_j R_j \left(\Sigma_{t\in[S_j,E_j]} b_j(t)\right) \quad (4a)$$

$$\text{s.t.} \Sigma_t b_j(t) \leq B_j, \quad \forall j \quad (4b)$$
$$es(t+1) = \rho(es(t) - e(t)), \quad \forall t \quad (4c)$$
$$0 \leq b_j(t) \leq MP_j, \quad \forall j, \forall t \quad (4d)$$
$$0 \leq d(t) \leq D, \quad \forall t \quad (4e)$$
$$0 \leq es(t) \leq ES. \quad \forall t \quad (4f)$$

Here $d(t)$ is given by (3). (4b) means the amount of served batch jobs cannot exceed the total demand, and could become $\Sigma_t b_j(t) = B_j$ if finishing all class j batch job is required. (4c) updates the energy storage level of each timeslot. We also incorporate constraints on maximum parallelization (4d), IT capacity (4e), and energy storage capacity (4f). We may have other constraints, such as a "net zero" constraint that the total energy consumed be less than the total renewable generation within $[1, T]$, i.e. $\sum_t (d(t) + c(d(t))) \leq \sum_t r(t)$. Note that, though highly detailed, this formulation does ignore some important concerns of data center design, e.g., reliability and availability. Such issues are beyond the scope of this paper; nevertheless, our designs merge nicely with proposals such as [32] for these goals.

In this paper, we restrict our focus from optimization (4a) to (5a), but the analysis can be easily extended to other convex cost functions.

$$\min_{\mathbf{b},\mathbf{e}} \Sigma_t p(t)(d(t) + c(d(t)) - r(t) - e(t))^+ - \Sigma_j R_j \left(\Sigma_{t\in[S_j,E_j]} b_j(t)\right) \quad (5a)$$

$$\text{s.t.} \Sigma_t b_j(t) \leq B_j, \quad \forall j \quad (5b)$$
$$es(t+1) = \rho(es(t) - e(t)), \quad \forall t \quad (5c)$$
$$0 \leq b_j(t) \leq MP_j, \quad \forall j, \forall t \quad (5d)$$
$$0 \leq d(t) \leq D, \quad \forall t \quad (5e)$$
$$0 \leq es(t) \leq ES. \quad \forall t \quad (5f)$$

Note that this optimization problem is jointly convex in $b_j(t)$ and $e(t)$ and can therefore be efficiently solved.

Given the significant amount of prior work approaching data center workload management via convex optimization [16, 17, 18, 19, 3, 11, 20], it is important to note the key difference between our formulation and prior work–our formulation is the first, to our knowledge, to incorporate renewable generation, storage, an optimized cooling micro grid, and batch job scheduling with consideration of both price diversity and temperature diversity. Prior formulations have included only one or two of these features. This "universal" inclusion is what allows us to consider truly integrated workload management.

3.5 Properties of the optimal workload management

The usage of the workload management optimization described above depends on more than just the ability to solve the optimization quickly. In particular, the solutions must be practical if they are to be adopted in actual data centers.

In this section, we provide characterizations of the optimal solutions to the workload management optimization, which highlight that the structure of the optimal solutions facilitates implementation. Specifically, one might worry that the optimal solutions require highly complex scheduling of

[2]Revenue is also derived from the interactive workload, but for the purposes of workload management the amount of revenue from this workload is fixed.

the batch jobs, which could be impractical. For example, if a plan schedules too many jobs at a time, it may not be practical because there is often an upper limit on how many workloads can be hosted in a physical server. The following results here show that such concerns are unwarranted.

Energy usage and cost

Although it is easily seen that the workload management optimization problem has at least one optimal solution,[3] in general, the optimal solution is not unique. Thus, one may worry that the optimal solutions might have very different properties with respect to energy usage and cost, which would make capacity planning difficult. However, it turns out that the optimal solution, though not unique, has nice properties with respect to energy usage and cost.

In particular, we prove that all optimal solutions use the same amount of power from the grid at all times. Thus, though the scheduling of batch jobs and the usage of energy storage might be very different, the aggregate grid power usage is always the same. This is a nice feature when considering capacity planning of the power system. Formally, this is summarized by the following theorem proved in [28].

Theorem 1. *For the simplified energy cost model (5a), suppose the optimal cooling power $c(d)$ is strictly convex in d. Then, the energy usage from the grid, $(d(t) + c(d(t)) - r(t) - e(t))^+$, at each time t is common across all optimal solutions.*

Though Theorem 1 considers a general setting, it is not general enough to include the optimal cooling substructure discussed in Section 3.1, which includes a strictly convex section followed by a linear section (while in practice, the chiller power is usually strictly convex in IT power, and satisfies the requirement of Theorem 1). However, for this setting, there is a slightly weaker result that still holds–the marginal cost of power during each timeslot is common across all optimal solutions. This is particularly useful because it then provides the data center operator a benchmark for evaluating which batch jobs are worthy of execution, i.e., provide revenue larger than the marginal cost they would incur. Formally, we have the following theorem proved in [28].

Theorem 2. *For the simplified energy cost model (5a), suppose $c(d)$ is given by (2). Then, the marginal cost of power, $\partial\left(p(t)(d(t) + c(d(t)) - r(t) - e(t))^+\right)/\partial(d(t))$, at each time t is common across all optimal solutions.*

Complexity of the schedule for batch jobs

A goal of this work is to develop an efficient, implementable solution. One practical consideration is the complexity of the schedule for batch jobs. Specifically, a schedule must not be too "fragmented", i.e., have many batch jobs being run at the same time and batch jobs being split across a large number of time slots. This is particularly important in virtualized server environments because we often need to allocate a large amount of memory for each virtual machine and the number of virtual machines sharing a server is often limited by the memory available to virtual machines even if the CPUs can be well shared. Additionally, there is always overhead associated with hosting virtual machines. If we run too many virtual machines on the same server at the same time, the CPU, memory, I/O and performance can be affected. Finally, with more virtual machines, live migrations and consolidations during runtime management can affect the system performance.

[3]This can be seen by applying Weierstrass' theorem [33], since the objective function is continuous and the feasible set is compact subset of $\mathbb{R}^n$.

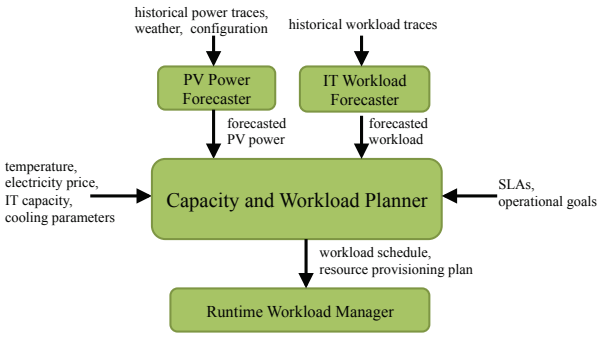


Figure 7: System Architecture

However, it turns out that one need not worry about an overly "fragmented" schedule, since there always exists a "simple" optimal schedule. Formally, we have the following theorem, which is proved in [28].

Theorem 3. *There exists an optimal solution to the workload management problem with at most $(T + J - 1)$ of the $b_j(t)$ are neither 0 nor MP_j.*

Informally, this result says that there is a simple optimal solution that uses at most $(T+J-1)$ more timeslots, or corresponding active virtual machines, in total than any other solutions finishing the same number of jobs. Thus, on average, for each class of batch job per timeslot, we run at most $\frac{(T+J-1)}{TJ}$ more active virtual machines than any other plan finishing the same amount of batch jobs. If the number of batch job classes is not large, on average, we run at most one more virtual machine per slot. In our experiments in Section 5, the simplest optimal solution only uses 4% more virtual machines. Though Theorem 3 does not guarantee that every optimal solution is simple, the proof is constructive. Thus, it provides an approach that allows one to transform an optimal solution into the simplest optimal solution.

In addition to Theorem 3, there are two other properties of the optimal solution that highlight its simplicity. We state these without proof due to space constraints. First, when multiple classes of batch jobs are served in the same timeslot, all of them except possibly the one with the lowest revenue are finished. Second, in every timeslot, the lowest marginal revenue of a batch job that is served is still larger than the marginal cost of power from Theorem 2.

4. SYSTEM PROTOTYPE

We have designed and implemented a supply-aware workload and capacity management prototype in a production data center based on the description in the previous section. The data center is equipped with on-site PV power generation and outside air cooling. The prototype includes workload and capacity planning, runtime workload scheduling and resource allocation, renewable generation and IT workload demand prediction. Figure 7 depicts the system architecture. The predictors use the historical traces of supply and demand information to predict the available power of an on-site PV, and the expected interactive workload demand. The capacity planner takes the predicted energy supply and cooling information as inputs and generates an optimal capacity allocation scheme for each workload. Finally, the runtime workload manager executes the workload plan.

The remainder of this section provides more details on each component of the system: capacity and workload planning, PV and workload prediction, and runtime workload management.

4.1 Capacity and Workload Planner

The data center has mixed energy sources: on-site photovoltaic (PV) generation tied to grid power. The cooling infrastructure has a cooling micro grid, including outside air cooling and chiller cooling. The data center hosts interactive applications and batch jobs. There are SLAs associated with the workloads. Though multiple IT resources can be used by IT workloads, we focus on CPU resource management in this implementation. We use a virtualized server environment where different workloads can share resources on the same physical servers.

The planner takes the following inputs: power supply (time varying PV power and grid power price data), interactive workload demand (time varying user request rates, response time target), batch job resource demands (CPU hours, arrival time, deadline, revenue of each batch job), IT configuration information (number of servers, server idle and peak power, capacity) and cooling configuration parameters (blower capacity, chiller cooling efficiency) and operational goals. We use the optimization (5a) in Section 3.4 with the following additional details.

We first determine the IT resource demand of interactive workload i using the M/GI/1/PS model, which gives $\frac{1}{\mu_i - \lambda_i(t)/a_i(t)} \leq rt_i$. Thus, the minimum CPU capacity needed is $a_i(t) = \frac{\lambda_i(t)}{\mu_i - 1/rt_i}$, which is a linear function of the arrival rate $\lambda_i(t)$. We estimate μ_i through real measurements and set the response time requirement rt_i according to the SLAs. While the model is not perfect for real-world data center workloads, it provides a good approximation. Although important, performance modeling is not the focus of this paper. The resulting average CPU utilization of interactive workload i is $1 - \frac{1}{\mu_i rt_i}$, therefore its actual CPU usage at time t is $a_i(t)\left(1 - \frac{1}{\mu_i rt_i}\right)$, the remaining $a_i(t)\frac{1}{\mu_i rt_i}$ capacity can be shared by batch jobs. For a batch job j, assume at time t it shares $n_{ji}(t) \geq 0$ CPU resource with interactive workload i and uses additional $n_j(t) \geq 0$ CPU resource by itself, then its total CPU usage at time t is $b_j(t) = \Sigma_i n_{ji}(t) + n_j(t)$, which is used to update Constraint (5b) and (5d). We have an additional constraint on CPU capacity that can be shared $\Sigma_j n_{ji}(t) \leq a_i(t)\frac{1}{\mu_i rt_i}$. Assume the data center has D CPU capacity in total, so the IT capacity constraint becomes $\Sigma_i a_i(t) + \Sigma_j n_j(t) \leq D$. Although our optimization (5a) in Section 3.4 can be used to handle IT workload with multi-dimensional demand, e.g., CPU, memory, here we restrict our attention to CPU-bound workloads.

The next step is to estimate the IT power consumption, which can be done based on the average CPU utilization

$$P_{server}(u) = P_i + (P_b - P_i) * u$$

where u is the average CPU utilization across all servers, P_i and P_b are the power consumed by the server at idle and their fully utilized state, respectively. This simple model has proved very useful and accurate in modeling power consumption since other components' activities are either static or correlate well with CPU activity [1]. Assuming each server has Q CPU capacity, using the above model, we estimate the IT power as follows[4]:

$$d(t) = \frac{\Sigma_i a_i(t)}{Q}\left(P_i + (P_b - P_i) * u_i\right) + \frac{\Sigma_j n_j(t)}{Q} P_b,$$

where $u_i = \left(1 - \frac{1}{\mu_i rt_i} + \frac{\Sigma_j n_{ji}(t)}{a_i(t)}\right)$.

The cooling power can be derived from the IT power according to the cooling model (2) described in Section 3.1.

[4]Since the number of servers used by an interactive workload or a class of batch jobs is usually large in data centers, we treat it as continuous.

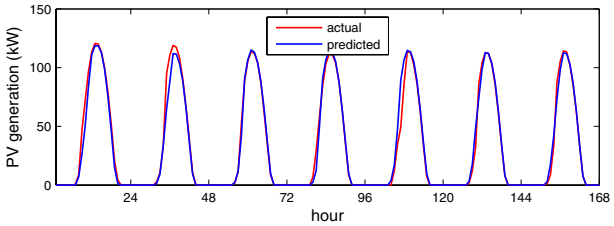


Figure 8: PV prediction

By solving the cost optimization problem (5a), we then obtain a detailed capacity plan, including at each time t the capacity allocated to each class j of batch jobs $b_j(t)$ (from $n_{ji}(t)$ and $n_j(t)$), and interactive workload i $a_i(t)$, energy storage usage $e(t)$, as well as optimal cooling configuration (i.e., capacity for outside air cooling and chiller cooling).

It follows from Section 3.4 that this problem is a convex optimization problem and hence there exist efficient algorithms to solve this problem. For example, disciplined convex programming [34] can be used. Under this approach, convex functions and sets are built up from a small set of rules from convex analysis, starting from a base library of convex functions and sets. Constraints and objectives that are expressed using these rules are automatically transformed to a canonical form and solved. In our prototype, the algorithm is implemented using Matlab CVX [34], a modeling system for convex optimization.

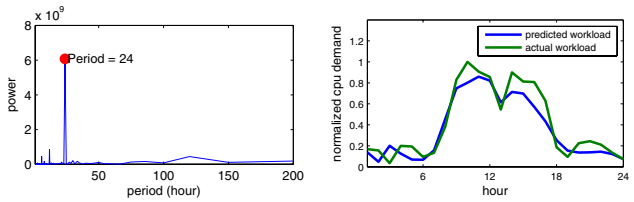
We then utilize the Best Fit Decreasing (BFD) method [35] to decide how to place and consolidate the workloads at each timeslot. More advanced techniques exist for optimizing the workload placement [5], but they are out this paper's scope.

4.2 PV Power Forecaster

A variety of methods have been used for fine-grained energy prediction, mostly using classical auto-regressive techniques [36, 37]. However, most of the work does not explicitly use the associated weather conditions as a basis for modeling. The work in [38] considered the impact of the weather conditions explicitly and used an SVM classifier in conjunction with a RBF kernel to predict solar irradiation. We use a similar approach for PV prediction in our prototype implementation. In order to predict PV power generation for the next day, we use a k-nearest neighbor (k-NN) based algorithm. The prediction is done at the granularity of one-hour time periods. The basic idea is to search for the most "similar" days in the recent past (using one week worked well here[5]) and use the generation during those days to estimate the generation for the target hour. The similarity between two days is determined using features such as ambient temperature, humidity, cloud cover, visibility, sunrise/sunset times on those days, etc. In particular, the algorithm uses weighted k-NN, where the PV prediction for hour t on the next day is given by $\hat{y}_t = \frac{\Sigma_{i \in N_k(\mathbf{x}_t, \mathcal{D})} y_i / d(\mathbf{x}_i, \mathbf{x}_t)}{\Sigma_{i \in N_k(\mathbf{x}_t, \mathcal{D})} 1/d(\mathbf{x}_i, \mathbf{x}_t)}$, where $\hat{y}_t$ is the PV predicted output at hour t, $\mathbf{x}_t$ is the feature vector, e.g., temperature, cloud cover, for the target hour t obtained from the weather forecast, y_i is the actual PV output for neighbor i, $\mathbf{x}_i$ is the corresponding feature vector, d is the distance metric function, $N_k(\mathbf{x}_i, \mathcal{D})$ are k-nearest neighbors of $\mathbf{x}_i$ in $\mathcal{D}$. k is chosen based on cross-validation of historical data.

Figure 8 shows the predicted and actual values for the PV supply of the data center for one week in September 2011. The average prediction errors vary from 5% to 20%. The prediction accuracy depends on occurrence of similar weather conditions in the recent past and the accuracy of the weather forecast. The results in [28] show that a ballpark

[5]If available, data from past years could also be used.



(a) FFT analysis (b) Workload prediction

Figure 9: Workload analysis and prediction

approximation is sufficient for planning purposes and our system can tolerate prediction errors in this range.

4.3 IT Workload Forecaster

In order to perform the planning, we need knowledge about the IT demand, both the stochastic properties of the interactive application and the total resource demand of batch jobs. Though there is large variability in workload demands, workloads often exhibit clear short-term and long-term patterns. To predict the resource demand (e.g., CPU resource) for interactive applications, we first perform a periodicity analysis of the historical workload traces to reveal the length of a pattern or a sequence of patterns that appear periodically. Fast Fourier Transform (FFT) can be used to find the periodogram of the time-series data. Figure 9(a) plots the time-series and the periodogram for 25 work days of a real CPU demand trace from an SAP application. From this we derive periods of the most prominent patterns or sequences of patterns. For this example, the peak at 24 hours in the periodogram indicates that it has a strong daily pattern (period of 24 hours). Actually, most interactive workloads exhibit prominent daily patterns. An auto-regressive model is then used to capture both the long term and short term patterns. The model estimates $w(d,t)$, the demand at time t on day d, based on the demand of the previous N days as $w(d,t) = \Sigma_{i=1}^{N} a_i * w(d-i,t) + c$. The parameters are calibrated using historical data.

We evaluate the workload prediction algorithm with several real demand traces. The results for a Web application trace are shown in Figure 9(b). The average prediction errors are around 20%. If we can use the previous M time points of the same day for the prediction, we could further reduce the error rate.

The total resource demand (e.g., CPU hours) of batch jobs can be obtained from users or from historical data or through offline benchmarking [39]. Like supply prediction, a ballpark approximation is good enough, as we will see in [28].

4.4 Runtime Workload Manager

The runtime workload manager schedules workloads and allocates CPU resource according to the plan generated by the planner. We implement a prototype in a KVM-based virtualized server environment [40]. Our current implementation uses a KVM/QEMU hypervisor along with control groups (Cgroups), a new Linux feature, to perform resource allocation and workload management [40]. In particular, it executes the following tasks according to the plan: (1) create and start virtual machines hosting batch jobs; (2) adjust the resource allocation (e.g., CPU shares or number of virtual CPUs) to each virtual machine; (3) migrate and consolidate virtual machines via live migration. The workload manager assigns a higher priority to virtual machines running interactive workloads than virtual machines for batch jobs via Cgroups. This guarantees that resources are available as needed by interactive applications, while excess resources can be used by the batch jobs, improving server utilization.

5. EVALUATION

To highlight the benefits of our design for renewable and cooling aware workload management, we perform a mixture of numerical simulations and experiments in a real testbed. We first present trace-based simulation results in Section 5.1, and then the experimental results on the real testbed implementation in Section 5.2.

5.1 Case Studies

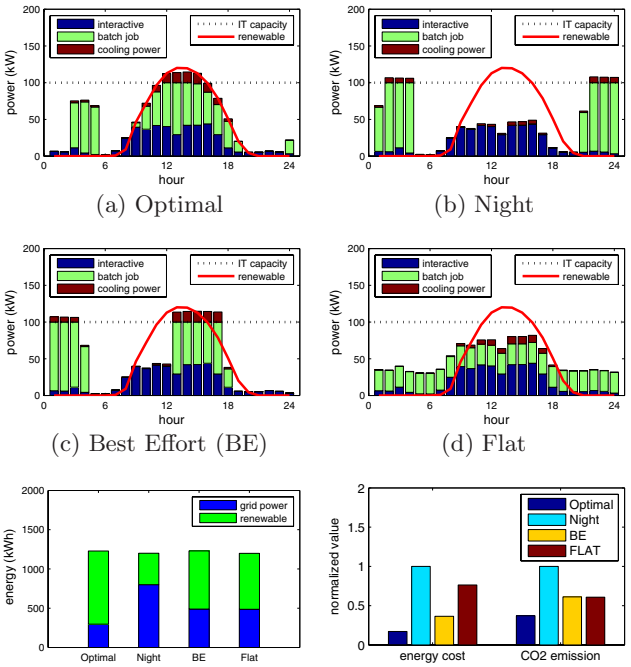
We begin by discussing evaluations of our workload and capacity planning using numerical simulations. We use traces from real data centers. In particular, we obtain PV supply, interactive IT workload, and cooling data from real data center traces. The renewable energy and cooling data is from measurements of a data center in California. The data center is equipped with 130kW PV panel array and a cooling system consisting of outside air cooling and chiller cooling. We use the real-time electricity price of the data center location obtained from [41]. The total IT capacity is 500 servers (100kW). The interactive workload is a popular web service application with more than 85 million registered users in 22 countries. The trace contains average CPU utilization and memory usage as recorded every 5 minutes. Additionally, we assume that there are a number of batch jobs. Half of them are submitted at midnight and another half are submitted around noon. The total demand ratio between the interactive workload and batch jobs is 1:1.5. The interactive workload is deemed critical and the resource demand must be met while the batch jobs can be rescheduled as long as they finish before their deadlines. The plan period is 24-hours and the capacity planner creates a plan for the next 24-hours at midnight based on renewable supply and cooling information as well as the interactive demand. The plan includes hourly capacity allocation for each workload. We assume perfect knowledge about the workload demand and renewable supply, and the results in [28] validate our solution works well with prediction errors and different mixes of interactive and batch workloads.

Here we explore: (i) How valuable is renewable and cooling aware workload management? (ii) Is net zero possible under renewable and cooling aware workload management? (iii) What portfolio of renewable sources is best?

How valuable is renewable and cooling aware workload management?

We start with the key question for this paper: how much energy cost/CO_2 savings does renewable and cooling aware workload management provide? In this study, we assume half of batch jobs must be finished before noon and another half must be finished before midnight. We compare the following four approaches: (i) *Optimal*, which integrates supply and cooling information and uses our optimization algorithm to schedule batch jobs; (ii) *Night*, which schedules batch jobs at night to avoid interfering with critical workloads and to take advantage of idle machines (this is a widely used solution in practice); (iii) *Best Effort (BE)*, which runs batch jobs immediately when they arrive and uses all available IT to finish batch jobs as quickly as possible; (iv) *Flat*, which runs batch jobs at a constant rate within the deadline period for the jobs.

Figure 10 shows the detailed schedule and power consumption for each approach, including IT power (batch and interactive workloads), cooling, and supply. As shown in the figure, compared with other approaches, *Optimal* reshapes the batch job demand and fully utilizes the renewable supply, and uses non-renewable energy, if necessary, to complete the batch jobs during this 24-hour period. These additional batch jobs are scheduled to run between 3am and 6am or 11pm and midnight, when the outside air cooling is most efficient and/or the electricity price is lower. As a result, our



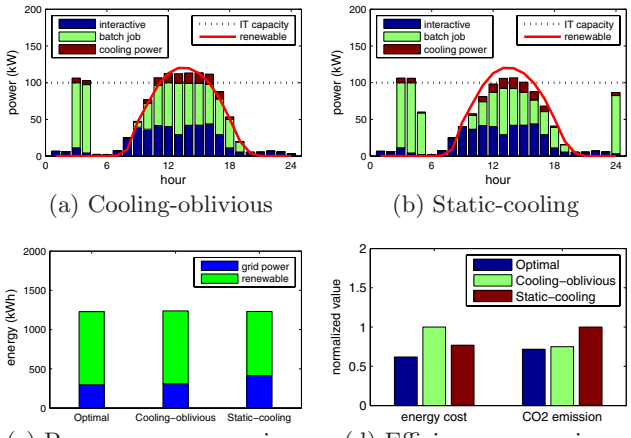
(a) Optimal (b) Night (c) Best Effort (BE) (d) Flat (e) Power usage comparisons (f) Efficiency comparisons

Figure 10: Power cost minimization while finishing all jobs

solution reduces the grid power consumption by 39%-63% compared to other approaches (Figure 10(e)). Though not clear in the figure, the optimal solution does consume a bit more total power (up to 2%) because of the low cooling efficiency around noon. Figure 10(f) shows the average recurring electricity cost and CO_2 emission per job. The energy cost per job is reduced by 53%-83% and the CO_2 emission per job is reduced by 39%-63% under the *Optimal* schedule. The adaptation of workload management to renewable availability is clear in Figure 10. Less clear is the importance of managing the workload in a manner that is "cooling aware".

The importance of cooling aware scheduling: As discussed in Sections 2.2 and 3.1, the cooling efficiency and capacity of a cooling supply often vary over time. This is particularly true for outside air cooling. One important component of our solution is to schedule workloads by taking into account time varying cooling efficiency and capacity. To understand the benefits of cooling integration, we compare our optimal solution as shown in Figure 10(a) with two solutions that are renewable aware but handle cooling differently: (i)*Cooling-oblivious* ignores cooling power and considers IT power only, (ii)*Static-cooling* uses a static cooling efficiency (assuming the cooling power is 30% of IT power) to estimate the cooling power from IT power and incorporates the cooling power into workload scheduling.

Figures 11(a) and 11(b) show *Cooling-oblivious* and *Static-cooling* schedules, respectively. As shown in the figures, both schedules integrate renewable energy into scheduling and run most batch jobs when renewables are available. However, because they do not capture the cooling power accurately, they cannot fully optimize workload schedule. In particular, by ignoring the cooling power, *Cooling-oblivious* underestimates the power demand and runs more jobs than the available PV supply and hence uses more grid power during the day. This is also less cost-efficient because the electricity price peaks at that time. On the other hand, by overestimating the cooling power demand, *Static-cooling* fails to fully utilize the renewable supply and results in inefficiency, too. Figures 11(c) and 11(d) compare the total power usage and energy efficiency,



(a) Cooling-oblivious (b) Static-cooling (c) Power usage comparisons (d) Efficiency comparisons

Figure 11: Benefit of cooling integration

i.e., normalized energy cost and carbon emission per batch job of these two approaches and *Optimal*, respectively. By accurately modeling the cooling efficiency and adapting to time variations in cooling efficiency, our solution reduces the energy cost by 20%-38% and CO_2 emission by 4%-28%.

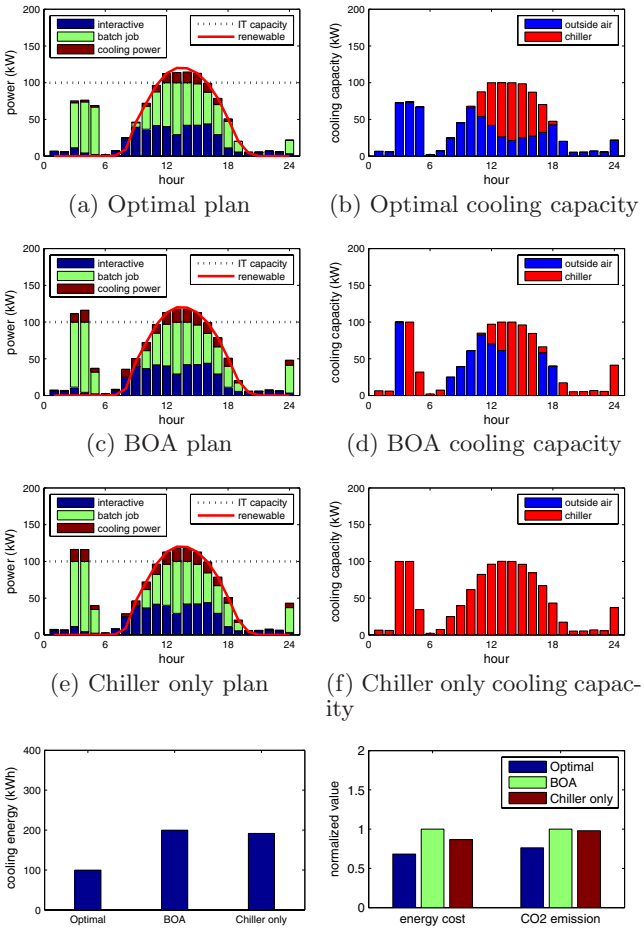
The importance of optimizing the cooling micro-grid: Optimizing the workload scheduling based on cooling efficiency is only one aspect of our cooling integration. Another important aspect is to optimize the cooling micro-grid, i.e, using the proper amount of cooling capacity from each cooling resource. This is important because different cooling supplies exhibit different cooling efficiencies as the IT demand and external conditions such as OAT changes. Our solution takes this into account and optimizes the cooling capacity management in addition to IT workload management.

We compare: (i) *Optimal* adjusts the cooling capacity for outside air cooling and chiller cooling based on the dynamic cooling efficiency, which is determined by IT demand and outside air temperature; (ii) *Binary Outside Air (BOA)* uses outside air cooling at its full capacity if OAT exceeds some threshold (25°C) or interactive demand is too low (less than 10% of the IT capacity) and does not use it at all otherwise; (iii) *Chiller only* uses the chiller cooling only. All three solutions are renewable and cooling aware and schedule workload according to the renewable supply and cooling efficiency. They finish the same number of batch jobs. The difference is how they manage cooling resources and capacity.

Figure 12 shows the cooling capacity from outside air cooling and chiller cooling for these three solutions. As shown in this figure, *Optimal* uses outside air only during night when it is more efficient, and combines outside air cooling and chiller cooling during other times. In particular, our solution uses less outside air cooling and more chiller cooling between 1pm and 4pm as outside air cooling is less efficient due to high IT demand and outside air temperature at that time. In contrast, *BOA* runs outside air at full capacity except the hours around noon when the outside air is less efficient. Figures 12(g) and 12(h) compare the power, cost, and cooling consumption of the three approaches. By optimizing the cooling substructure, our solution reduces the cooling power by 66% over *BOA* and 48% over *Chiller only*.

Is net zero energy consumption possible with renewable and cooling aware workload management?

Now, we switch our goal from minimizing the cost incurred by the data center to minimizing the environmental impact of the data center. Net zero is often used to describe a building with zero net energy consumption and zero carbon emission annually. Recently, researchers have envisioned how net zero



(a) Optimal plan (b) Optimal cooling capacity

(c) BOA plan (d) BOA cooling capacity

(e) Chiller only plan (f) Chiller only cooling capacity

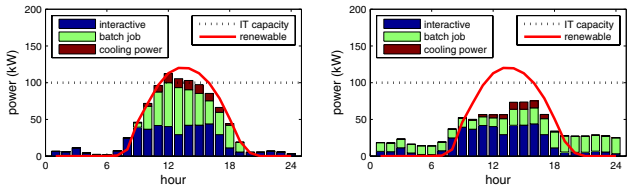
(g) Cooling power (h) Efficiency comparisons

Figure 12: Benefit of cooling optimization

building concepts can be effectively extended into the data center space to create a net zero data center, whose total power consumption is less than or equal to the total power supply from renewable. We explore if net zero is possible with renewable and cooling aware workload management in data centers and how much it will cost.

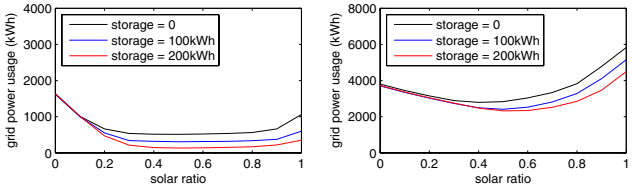
By adding a net-zero constraint (i.e., total power consumption $\leq$ total renewable supply) to our optimization problem, our capacity planner can generate a net-zero schedule. Figure 13(a) shows our solution (*Net-zero1*) for achieving a net zero operation goal. Similar to the optimal solution shown in Figure 10(a), *Net-zero1* optimally schedules batch jobs to take advantage of the renewable supply; however, batch jobs are only executed when renewable energy is available and without exceeding the total renewable generation, and thus some are allowed to not finish during this 24 hour period. In this case, about 40% of the batch jobs are delayed till a future time when a renewable energy surplus may exist. Additionally, some renewable energy is reserved to offset non-renewable energy used at night for interactive workloads. This excess renewable power is reserved in the afternoon when the outside air temperature peaks, thus minimizing the energy required for cooling.

A key to achieving net zero is energy storage. By maximizing renewable usage directly, *Net-zero1* reduces the dependency on storage and hence the capital cost. To understand the benefit, we compare *Net-zero1* with another schedule, *Net-zero2*, which runs the same amount of batch jobs but distributes the batch jobs over 24-hours as shown in Figure 13(b). Both approaches achieve the net-zero goal, but *Net-zero2* uses 287% more grid power compared to our solution *Net-zero1*. As a result, the energy storage sizes of *Net-zero1* and *Net-zero2* are 82kWh and 330kWh, respectively. Using an estimated cost of 400$/kWh [22], this difference in energy demand results in $99,200 more energy storage expenditure for *Net-zero2*.



(a) Net-zero1 (b) Net-zero2

Figure 13: Net Zero Energy



(a) With turning off servers (b) Without turning off servers

Figure 14: Optimal renewable portfolio

What portfolio of renewable sources is best?

To this point, we have focused on PV solar as the sole source of renewable energy. Since wind energy is becoming increasingly cost-effective, a sustainable data center will likely use both solar and wind to some degree. The question that emerges is which one is better source from the perspective of optimizing data center energy efficiency. More generally, what is the optimal portfolio of renewable sources?

We conduct a study using the wind and solar traces depicted in Figure 2. Assuming an average renewable supply of 200kW, we vary the mix of solar and wind in the renewable supply. For each mix, we use our capacity management optimization algorithm to generate an optimal workload schedule. We compare the non-renewable power consumption for different renewable mixes for two cases (turning off unused servers and without turning off unused servers). Figure 14 shows the results as a function of percentage of solar with different storage capacity. As shown in the figure, the optimal portfolio contains more solar than wind because solar is less volatile and the supply aligns better with IT demand.

However, wind energy is still an important component and a small percentage of wind can help improve the efficiency. For example, the optimal portfolio without storage consists of about 60% solar and 40% wind. When we do not turn off unused servers, wind becomes more valuable due to the significant power consumption during night even when the server utilization is low.

In summary, solar is a better source for local data centers in sunny areas such like Palo Alto and a small addition of wind can help improve energy efficiency. The optimal portfolio varies for different areas. Additionally, recent work has shown that the value of wind increases significantly when geographically diverse data centers are considered [17, 20].

5.2 Experimental Results on a Real Testbed

The case studies described in the previous section highlight the theoretical benefits of our approach over existing solutions. To verify our claims and ensure that we have a

practical and robust solution, we experimentally evaluate our prototype implementation on a real data center testbed and contrast it with a current workload management approach.

5.2.1 Experiment Setup

Our testbed consists of four high end servers (each with two 12-core 1.8GHz processors and 64 GB memory) and the following workloads: one interactive Web application, and 6 batch applications. Each server is running Scientific Linux. Each workload is running inside a KVM virtual machine. The interactive application is a single-tier web application running multiple Apache web servers and batch jobs are sysbench [42] with different resource demands. httperf [43] is used to replay the workload demand traces in the form of CGI requests, and each request is directed to one of the Web servers. The PV, cooling data, and interactive workload traces used in the case study are scaled to the testbed capacity. We measure the power consumption via the server's built-in management interfaces, collect CPU consumption through system monitoring and obtain response times of Web servers from Apache logs.

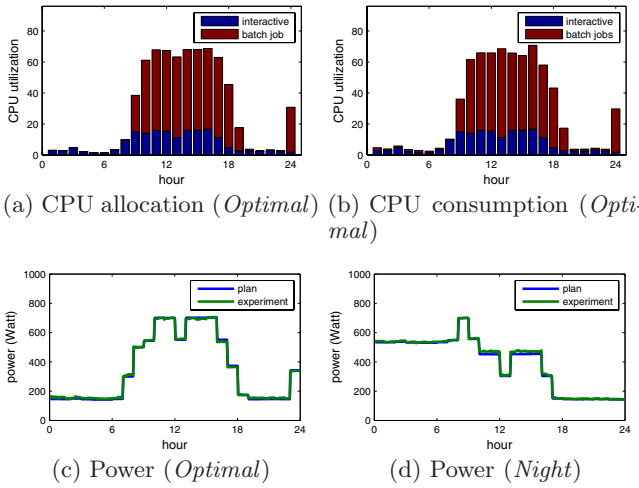
5.2.2 Experiment Results

We compare two approaches: (i) *Optimal* is our optimal design, (ii) *Night*, which runs batch jobs at night. For each plan, the runtime workload manager dynamically starts the batch jobs, allocates resources to both interactive and batch workloads and turns on/off servers according to the plan. Figures 15(a) and 15(b) shows the CPU allocation of our optimal solution and the actual CPU consumption measured in the experiment, respectively. The results show that actual resource usage closely follows the power generated by the capacity planner. The results for *Night* are similar. We further compare the predicted power usage in the plan and the actual power consumption in the experiment for both approaches. From Figures 15(c) and 15(d), we see that the actual power usage is close to the power plan.

We then compare the power consumption and performance of the two approaches. Figure 16(a) shows the power consumption. *Optimal* does more work during the day when the renewable energy source is available. *Night* uses additional servers from midnight to 6am to run batch jobs while our solution starts batch jobs around noon by taking advantage of renewable energy. Compared with *Night*, our approach reduces the grid power usage by 48%. One thing worth mentioning is that the total power is not quite proportional to the total CPU utilization as a result of the large idle part of the server power. This is most noticeable when the number of servers is small, as we see in the experimental results, Figures 15(b) and 15(c). When the total number of servers increases, the impact of idle power decreases and *Optimal* will save even more grid power.

One reason that batch jobs are scheduled to run at night is to avoid interfering with interactive workloads. Our approach runs more jobs during the day when the web server demand is high. To understand the impact of this on performance, we compare the average response time of the web server. The results show that both approaches are almost identical: 153.0ms for *Optimal*, compared to 156.0ms for *Night*. See [28] for more details. This is because both solutions satisfy web server demand and Linux KVM/Cgroups scheduler is preemptive and enables CPU resources to be effectively virtualized and shared among virtual machines [40]. In particular, assigning a much higher priority to virtual machines hosting the web servers guarantees that resources are available as needed by the web servers.

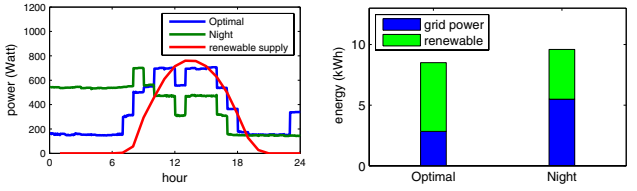
In summary, the real experiment results demonstrate that (i) the optimization-based workload management scheme can translate effectively into a prototype implementation, (ii) compared with traditional workload management solution,



(a) CPU allocation (*Optimal*) (b) CPU consumption (*Optimal*)

(c) Power (*Optimal*) (d) Power (*Night*)

Figure 15: Comparison of plan and experimental results



(a) Power consumption (b) Total power consumption

Figure 16: Comparison of optimal and night

Optimal significantly reduces the use of grid power without degrading the performance of critical demand.

6. CONCLUDING REMARKS

Our goal in this paper is to provide an integrated workload management system for data centers that takes advantage of the efficiency gains possible by shifting demand in a way that exploits time variations in electricity price, the availability of renewable energy, and the efficiency of cooling. There are two key points we would like to highlight about our design.

First, a key feature of the design is the integration of the three main data center silos: cooling, power, and IT. Though a considerable amount of work exists in optimizing efficiencies of these individually, there is little work that provides an integrated solution for all three. Our case studies illustrate that the potential gains from an integrated approach are significant. Additionally, our prototype illustrates that these gains are attainable. In both cases, we have taken case to measurements from a real data center and traces of real applications to ensure that our experiments are meaningful.

Second, it is important to point out that our approach uses a mix of implementation, modeling, and theory. At the core of our design is a cost optimization that is solved by the workload manager. Care has been taken in designing and solving this optimization so that the solution is "practical" (see the characterization theorems in Section 3.5). Building an implementation around this optimization requires significant measurement and modeling of the cooling substructure, and the incorporation of predictors for workload demand and PV supply. We see one role of this paper as a proof of concept for the wide-variety of "optimization-based designs" recently proposed, e.g., [16, 17, 18, 19, 3, 11, 20].

There are a number of future directions building on this work including integrating reliability and a more detailed study of the role of storage. But, perhaps the most exciting

direction is the potential to consider renewable and cooling aware workload management of geographically diverse data centers as opposed to the local workload management considered here. As illustrated in [17, 20], geographical diversity can be a significant aid in handling the intermittency of renewable sources and electricity price fluctuations. For example, the fact that fluctuations in wind energy are nearly uncorrelated for significantly distant locations means that wind can provide a nearly constant baseline supply of energy if workload can be adapted geographically. Another future direction we would like to highlight is to understand the impact of workload management on the capacity planning of a data center, e.g. the size of renewable infrastructure, the capacity of IT and cooling infrastructure, and the capital expense and to minimize the Total Cost of Ownership (TCO).

Acknowledgements

We are grateful to many members of Sustainable Ecosystem Research Group at HP Labs. Chandrakant Patel has provided great support and guidance at various stages of this work. Niru Kumari provided valuable information on chiller cooling models. Martin Arlitt, Amip Shah, Sergey Blagodurov and Alan McReynolds offered helpful feedback. We also thank the anonymous reviewers and our shepherd, Christopher Stewart, for their valuable comments and help.

7. REFERENCES

[1] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proc. of ACM ISCA*, 2007.
[2] A. Gandhi, M. Harchol-Balter, and C. L. R. Das, "Optimal power allocation in server farms," in *Proc. of ACM Sigmetrics*, 2009.
[3] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," in *Proc. of INFOCOM*, 2011.
[4] A. Wierman, L. L. H. Andrew, and A. Tang, "Power-aware speed scaling in processor sharing systems," in *Proc. of INFOCOM*, 2009.
[5] J. Choi, S. Govindan, B. Urgaonkar, and A. Sivasubramaniam, "Power consumption prediction and power-aware packing in consolidated environments," *IEEE Transactions on Computers*, vol. 59, no. 12, 2010.
[6] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No "power" struggles: Coordinated multi-level power management for the data center," in *Proc. of ASPLOS*, 2008.
[7] Z. Wang, A. McReynolds, C. Felix, C. Bash, and C. Hoover, "Kratos: Automated management of cooling capacity in data centers with adaptive vent tiles," in *Proc. of IMECE*, 2009.
[8] C. E. Bash, C. D. Patel, , and R. K. Sharma, "Dynamic thermal management of aircooled data centers," in *Proc. of ITHERM*, 2006.
[9] W. Huang, M. Allen-Ware, J. Carter, E. Elnozahy, H. Hamann, T. Keller, C. Lefurgy, J. Li, K. Rajamani, and J. Rubio, "Tapo: Thermal-aware power optimization techniques for servers and data centers," in *Proc. of IGCC*, 2011.
[10] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making scheduling cool: Temperature-aware workload placement in data centers," in *Proc. of USENIX ATC*, 2005.
[11] E. Pakbaznia and M. Pedram, "Minimizing data center cooling and server power costs," in *Proc. of ISLPED*, 2009.
[12] Y. Chen, D. Gmach, C. Hyser, Z. Wang, C. Bash, C. Hoover, and S. Singhal, "Integrated management of application performance, power and cooling in data centers," in *Proc. of NOMS*, 2010.
[13] T. Breen, E. Walsh, J. Punch, C. Bash, and A. Shah, "From chip to cooling tower data center modeling: Influence of server inlet temperature and temperature rise across cabinet," *Journal of Electronic Packaging*, vol. 133, no. 1, 2011.
[14] D. Gmach, J. Rolia, C. Bash, Y. Chen, T. Christian, A. Shah, R. Sharma, and Z. Wang, "Capacity planning and power management to exploit sustainable energy," in *Proc. of CNSM*, 2010.
[15] "Clean urban energy: Turn buildings into batteries." 2011.
[16] K. Le, O. Bilgir, R. Bianchini, M. Martonosi, and T. D. Nguyen, "Capping the brown energy consumption of internet services at low cost," in *Proc. IGCC*, 2010.
[17] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. H. Andrew, "Greening geographical load balancing," in *Proc. ACM Sigmetrics*, 2011.
[18] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment," in *Proc. of INFOCOM*, 2010.
[19] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, "Donar: decentralized server selection for cloud services," in *Proc. of ACM Sigcomm*, 2010.
[20] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. H. Andrew, "Geographical load balancing with renewables," in *Proc. ACM GreenMetrics*, 2011.
[21] M. Lin, Z. Liu, A. Wierman, and L. L. Andrew, "Online algorithms for geographical load balancing." http://http://www.cs.caltech.edu/ zliu2/onlineGLB.pdf.
[22] *Electricity Energy Storage Technology Options: A White Paper Primer on Applications, Costs and Benefits*. 2010.
[23] "Report to congress on server and data center energy efficiency." 2007.
[24] R. Zhou, Z. Wang, A. McReynolds, C. Bash, T. Christian, and R. Shih, "Optimization and control of cooling microgrids for data centers," in *Proc. of ITherm*, 2012.
[25] C. Patel, R. Sharma, C. Bash, and A. Beitelmal, "Energy flow in the information technology stack," in *Proc. of IMECE*, 2006.
[26] F. Bleier, *Fan Handbook: Selection, Application and Design*. New York: McGraw-Hill, 1997.
[27] J. Holman, *Heat Transfer. 8th ed*. New York: McGraw-Hill, 1997.
[28] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," Tech. Rep. HPL-2012-73, HP Labs, April 2012.
[29] R. Urgaonkar, B. Urgaonkar, M. Neely, and A. Sivasubramaniam, "Optimal power cost management using stored energy in data centers," in *Proc. of ACM Sigmetrics*, 2011.
[30] S. Ong, P. Denholm, and E. Doris, "The impacts of commercial electric utility rate structure elements on the economics of photovoltaic systems," Tech. Rep. NREL/TP-6A2-46782, National Renewable Energy Laboratory, 2010.
[31] L. L. H. Andrew, M. Lin, and A. Wierman, "Optimality, fairness and robustness in speed scaling designs," in *Proc. of ACM Sigmetrics*, 2010.
[32] E. Thereska, A. Donnelly, and D. Narayanan, "Sierra: a power-proportional, distributed storage system," Tech. Rep. MSR-TR-2009-153, Microsoft Research, 2009.
[33] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.
[34] http://cvxr.com/cvx/
[35] V. Vazirani, *Approximation Algorithms*. Springer, 2003.
[36] D. R. Cox, "Prediction by exponentially weighted moving averages and related methods," 1961.
[37] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," 2007.
[38] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," in *Proc. of SmartGridComm*, 2011.
[39] Y. Becerra, D. Carrera, and E. Ayguade, "Batch job profiling and adaptive profile enforcement for virtualized environments," in *Proc. of ICPDNP*, 2009.
[40] "Kvm kernel based virtual machine." http://www.redhat.com/f/pdf/rhev/DOC-KVM.pdf.
[41] http://www.ferc.gov/market-oversight/mkt electric
[42] "Sysbench: a system performance benchmark." http://sysbench.sourceforge.net/.
[43] http://www.hpl.hp.com/research/linux/httperf/