Integrating Renewable Energy Using Data Analytics Systems: Challenges and Opportunities

Andrew Krioukov, Christoph Goebel[†], Sara Alspaugh, Yanpei Chen, David Culler, Randy Katz
Department of Electrical Engineering and Computer Science
University of California, Berkeley
International Computer Science Institute[†]
{krioukov,alspaugh,ychen2,culler,randy}@eecs.berkeley.edu
goebel@icsi.berkeley.edu

Abstract

The variable and intermittent nature of many renewable energy sources makes integrating them into the electric grid challenging and limits their penetration. The current grid requires expensive, large-scale energy storage and peaker plants to match such supplies to conventional loads. We present an alternative solution, in which supply-following loads adjust their power consumption to match the available renewable energy supply. We show Internet data centers running batched, data analytic workloads are well suited to be such supply-following loads. They are large energy consumers, highly instrumented, agile, and contain much scheduling slack in their workloads. We explore the problem of scheduling the workload to align with the time-varying available wind power. Using simulations driven by real life batch workloads and wind power traces, we demonstrate that simple, supply-following job schedulers yield 40-60% better renewable energy penetration than supply-oblivious schedulers.

1 Introduction

A major challenge for the future electric grid is to integrate renewable power sources such as wind and solar [26]. Such sources are variable and intermittent, unlike traditional sources that provide a controllable, steady stream of power. Integrating a substantial fraction of renewable sources into the energy mix typically requires extensive backup generation or energy storage capacity to remove the variable and intermittent nature of such sources [11]. Given technological and economic limitations in current energy storage techniques, it will be difficult to meet even the current mandates for renewable energy integration [6, 18, 26].

Some have proposed creating *supply-following electric loads* from home appliances, lighting, and electric vehicles [5, 10]. This approach would schedule or sculpt the electric load such that it is synchronized with power availability from renewable sources, e.g., charge electric vehicles only when sufficient wind or solar power is available. This dispatchable demand approach represents an advance over traditional demand response techniques, which focus only on shedding load during times of high demand. However, home appliances, lighting, and electric vehicles all directly interact with humans. Such human dependencies can limit when, how

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

much, and how quickly such loads can be re-scheduled or sculpted. Subjective aspects of human comfort and perception can make it challenging to quantify and to compare alternate systems.

Recent green computing efforts have addressed components of a solution to this problem: energy efficiency [2, 8, 13, 15, 27], power proportionality [4, 14, 17, 21, 25], and service migration to geographic areas of lower real-time electricity prices [19]. These efforts are only components because even if we have energy efficient, power proportional systems that minimize energy bills, we will still have the problem of matching variable and intermittent energy sources with so-far less variable and continuous energy demand.

However, we show natural extensions of these techniques that address the matching problem on data analytics computer clusters. These clusters exhibit several properties. First, such clusters have varying levels of utilization [4], with the serviced workload having significant scheduling slack [10]. Second, the automatic and batch processing nature of computations on these clusters partially remove human limitations on when and how much the workload can be re-scheduled or sculpted. Third, the highly engineered and networked nature of such clusters allow rapid response to control signals from renewable sources. Taken together, these properties make data analytics computer clusters a compelling building block for supply-following loads.

This paper shows how to build supply-following loads using data analytics computer clusters.

- We make the case that data analytics workloads presenting a unique opportunity to implement supply-following mechanisms to help address the problem of integrating renewable energy.
- We introduce a quantitative metric to measure the degree renewable energy integration.
- We describe a simple supply-following job scheduler, evaluate it using realistic wind power and data analytic workload traces, and attain 40-60% improvement in the level of renewable energy integration.

The rest of the paper is organized as follows. Section 2 surveys the technical landscape to explain why the techniques we present are not in use today. Section 3 formalizes the problem of integrating renewable energy and introduces a metric for quantifying the degree of integration. Section 4 describes our simulation-based methodology, and the particular wind power traces and data analytic workloads we considered. Section 5 describes our supply-following scheduling algorithms. Section 6 presents the results of our simulations, which show that our algorithm yields significant improvement in renewable energy integration. Lastly, we discuss in Section 7 the key opportunities and challenges for future research in the area.

2 Technical Landscape

The intermittent and variable nature of renewable sources of energy, such as wind and solar, pose a problem for electric grid operators, who face increasing pressure to enlarge their renewable generation capacity. The current model of electric grid operation predicts the load in advance and then schedules the supply portfolio to service the load. The baseline generation capacity comes from sources that output constant, relatively inexpensive power, such as large coal and nuclear power plants. A portfolio of smaller, rapid-response, but more expensive and intermittent peaker plants track variation in demand and bridge any transient discrepancies between predicted and actual loads. This represents a model of *load-following supplies*, in which the electric loads are oblivious to the amount or type of supply, and supplies must track the electric load. Increasing the proportion of renewable supplies severely disrupts this model because renewable sources simply cannot be scheduled on demand.

One approach is to compensate for the variance in renewable supply using energy storage or additional peaker plants. This is an expensive proposition using current technologies - the energy storage and peaker plants must meet the full peak-to-zero swing in supply, instead of just meeting the small gap between predicted and actual load. An alternate solution is to flip the relationship and schedule the loads, thus creating *supply-following loads*. In this approach, loads must be prepared to consume electricity when supply is available and not otherwise. Only some loads form appropriate building blocks for supply-following loads.

Data analytics clusters represent a good example of electricity consumers with inherent scheduling flexibility in their workload. In a data analytics or batch processing cluster, users submit jobs in a non-interactive fashion. Unlike interactive web service clusters, these clusters do not have short, strict deadlines for servicing submitted jobs. Job completion deadlines typically create slack for a scheduler to shift the workload in time and consequently adjust energy consumption to, for instance, the amount of renewable energy available, or when electricity is cheaper.

If this is the case, why aren't such techniques in common practice? Part of the answer is that green computing remains an emerging field, with existing research focused on "low-hanging-fruits". Only recently has renewable energy integration been recognized as an unsolved problem. We briefly illustrate this transition in research focus. Early efforts in green computing included the Power Utilization Efficiency (PUE) of large scale data centers. PUE is defined as the ratio of total data center consumption to that consumed by the computing equipment, with typical values of 2 or greater [9, 24], i.e., to deliver 1 unit of energy to the computers, the data center wastes 1 or more units of energy in the power distribution and cooling infrastructure [3]. This revealed huge inefficiencies in the physical designs of data centers, and intense design efforts removed this overhead and reduced PUE to 1.2-1.4, much close to the ideal value of 1.0 [20, 22].

Once PUE values became more acceptable, data center operators recognized that real measure of effectiveness is not the power ratio between servers and the power distribution/cooling facilities, but the actual work accomplished on the servers per unit energy. In fact, servers in data centers are actively doing work typically only about 25% of the time [4]. Such low utilization levels naturally follow from the gap between peak and average requests rates, amplified by overprovisioning to accommodate transient workload bursts. Consequently, data center designers identified the need for *power proportionality*, i.e., that systems should consume power proportional to the dynamically serviced load and not to the static overprovisioning [4, 14, 17, 21, 25].

Power proportionality is a prerequisite for successfully turning data analytics clusters into supply-following loads. Otherwise, the cluster consumes approximately the same amount of energy regardless of the work it is doing. Unfortunately, modern server platforms are far from power proportional despite substantial improvements in power efficiency of the microprocessor, including Dynamic Voltage/Frequency Scaling (DVFS) and the introduction of a family of sophisticated power states. Even for specially engineered platforms [4], the power consumed when completely idle is over 50% of that when fully active, and idle consumption often over 80% of peak for commodity products [7].

Recently, we demonstrated the design and implementation of power proportional clustered services constructed out of non-power proportional systems. The basic approach is fairly obvious – put idle servers to sleep and wake them up when they are needed, keeping just enough extra active capacity to cover the time to respond to changes [12]. Thus, the stage is set for creating supply-following loads from data analytic compute clusters.

3 Problem Formulation

Our high-level goal is to use increase renewable energy use by turning data analytics clusters into supply-following electric loads. We consider a specific scenario where are data centers located near sources of clean electricity seek to maximize the use of local, directly attached wind turbines (or solar panels). In addition to the local intermittent power source, we can also draw energy from traditional sources in the grid. We observe the available renewable power at a given time and respond accordingly by sculpting the data analytics workload. If our data center is truly supply-following, it would draw most of its energy from the local, directly attached renewably supply, and very little energy from the rest of the grid.

> Key idea: Measure the degree of renewable integration by the fraction of total energy that comes from the renewable source, i.e., wind energy used divided by the total energy used. Better wind integration corresponds to a higher percentage.

Alternate problem formulations include optimizing a grid supply "blend" using remote control signals from grid operators, or responding to real time energy price, with the price being a function of the renewable and conventional power blend. These formulations assume that renewable sources have already been integrated in the grid signaling/pricing structures, and complicates validating the quality of such integration. Thus, we choose the strict formulation in which the data center operators directly contribute quantifiable improvements in integrating renewable sources.

A key feature of data analytics clusters is that jobs often do not need to be executed immediately. We use the term *slack* to describe the leeway that allows computational loads to be shifted in time. Slack is the number of time units that a job can be delayed, i.e., the slack for job j with submission time b_j, deadline d_j, that executes for t_j units of time is $s_j = d_j - b_j - t_j$.

Slack allows scheduling mechanisms to align job execution with the highly variable renewable power supplies. The quality of alignment, measured by the ratio of renewable to total energy used, depends on both the slack in the data analytic workload and the variability in the available renewable power. To obtain realistic results, we used batch job workload from a natural language processing cluster at UC Berkeley (Section 4.1), and wind power traces from the National Renewable Energy Laboratory (NREL) (Section 4.2).

We make several simplifying assumptions. We assume the cluster is power proportional. Otherwise, the cluster consumes roughly the same power all the time, making it incompatible with variable and intermittent sources. Also, we consider only data analytics applications that are inelastic, i.e., they cannot adjust the amount of consumed resources at runtime. An example of inelastic application is Torque [23], and an example of elastic applications is Hadoop [1]. Further, the application is "interruptible", meaning it can stop and resume as needed. At job submission time, we know the job deadline, run time, and resource requirements. We also assume that all the data needed by the application resides on a SAN that is under separate power management – it remains an open question to effectively power manage systems that co-locate computation and storage.

Slack is a key enabler for supply-following scheduling algorithms, in conjunction with power proportionality. Unlike traditional batch schedulers that try to maximize job throughput or minimize response time, supply-following schedulers seeks to find a good tradeoff between throughput, response time, and running jobs only when renewable energy is available.

4 Methodology

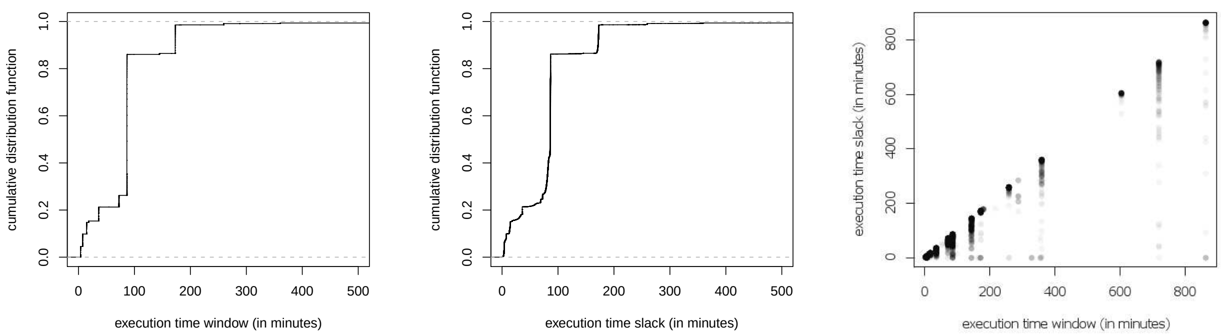
Two key components of our evaluation of supply-following scheduling are the input cluster workload and the input wind power traces. The degree of renewable integration depends on the slack in the particular cluster workload and the ability of the workload to align with particular wind traces.

4.1 Data Analytics Traces

We use batch job traces collected from a natural language processing cluster of 576 servers at UC Berkeley. Natural language processing involves CPU-intensive signal processing and model fitting computations. These jobs execute in a parallelized and distributed fashion on many processors. The completion deadlines are rarely critical. The cluster job management system is Torque [23], a widely used, open source resource manager providing control over batch jobs and distributed compute nodes. When submitting jobs to Torque, users specify the amount of processors and memory to be allocated, as well as the maximum running time. During job execution, the scheduler keeps track of the remaining running time of each job.

We collect job execution traces using Torque's `showq` command to sample the cluster state at 1 minute intervals. We collected a one month trace of 128,914 jobs and extracted job start times, end times and user-specified maximum running times. Deadlines are defined as the start time plus the maximum running time.

Figure 1(a) shows the CDF of the extracted job execution times. Figure 1(b) provides the CDF of execution



(a) CDF of execution times (b) CDF of execution time slack (c) Execution time window versus slack

Figure 1: Characteristics of batch job traces

time slack. The CDF shows that most of the jobs extracted from the cluster logs have a significant amount of execution time slack, generally ranging from 40 to 80 minutes of slack. Figure 1(c) shows the joint distribution of the job execution times and the execution time slack. The plot shows accumulations at certain execution time intervals (vertical lines), indicating different amounts of slack associated with jobs with the same execution times.

4.2 Wind Traces

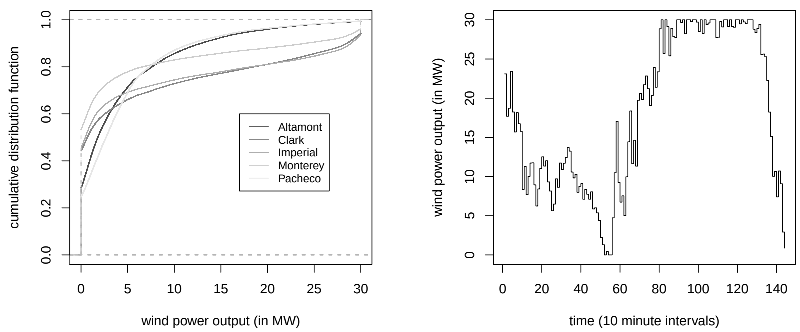
We used the wind speed and power data from the National Renewable Energy Laboratory (NREL) database [16]. This database contains time series data in 10 minute intervals from more than 30,000 measurement points in the Western Interconnection, which includes California. The measurement points in the NREL database are wind farms that hold 30 MW of installed capacity each. This capacity roughly equals 10 Vestas V-90 3MW wind turbines. For our experiments we picked one measurement point out of each major wind region in California.

Using wind output data from different regions is equivalent to considering data centers located in near different wind supplies. Our intention is to evaluate how well the the supply-following schedulers perform in range of possible locations. Figure 2(a) shows the cumulative distribution functions of wind power output at the different sites, suggesting considerable variation. Interestingly, some regions, such as Monterey, exhibit no power generation at all for large fractions of the time. Zero wind power generation results from either no wind or heavy storms causing the turbines to shut down.

Figure 2(b) shows the wind power output of a single wind farm during one day in the Altamont region. Wind power production can decline from maximum to zero output quickly, as indicated by the power drop at the right of the graph. Such steep rises and declines occur often in the traces. These fast transitions are arguably too short for re-scheduling human-facing loads such as home appliances and lighting.

4.3 Simulation Setup

The simulation takes as input the job submission times, job deadlines, required number of processors, and wind power availability over time. The simulation runs the candidate scheduler, and outputs the job execution order and power consumed over time. From these outputs, we then compute the percentage of total energy consumed that comes from wind. For the results in this paper, we run the simulation using one month of cluster jobs and wind power traces.



(a) CDFs of wind power output (b) Altamont wind power output over time

Figure 2: Characteristics of wind traces

5 Algorithms

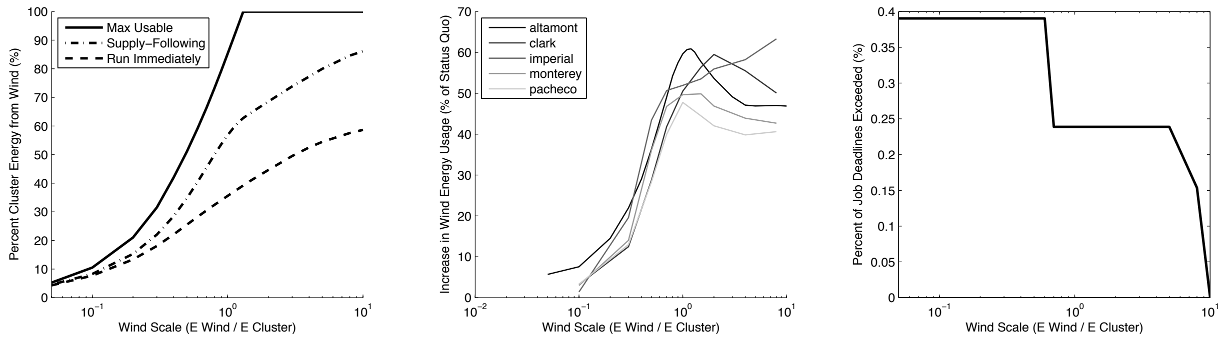
We compare two scheduling algorithms. The supply-oblivious, *run-immediately algorithm* executes jobs as soon as enough processors become available. Jobs that do not complete by their deadline are killed. The run-immediately algorithm represents the default scheduling behavior of Torque.

The *supply-following algorithm* attempts to align power consumption with the amount of wind power available, while minimizing the amount time by which jobs exceed their deadlines. It makes scheduling decisions at regular time intervals. At each interval, it schedules jobs that require immediate execution, beginning with jobs that have exceeded their deadlines the most, through jobs that will exceed their deadlines in the next time interval if left idled. If there are no such jobs that need immediate execution, the scheduler checks the wind power level. If some wind power is available, the scheduler executes the remaining jobs in order of increasing remaining slack, until either wind power or processors are fully used, or there are no more jobs on queue.

We use the heuristic of scheduling jobs in order of increasing slack, since jobs with a lot of slack can wait longer until more wind power becomes available. Thus, in the absence of accurate wind power or cluster workload predictors, this execution order increases the likelihood that we exploit all the available slack to align cluster workload and wind power availability.

One complication is that deferring jobs with slack can potentially aggravate resource bottlenecks. For example, if all jobs on the queue have slack and no wind power is available, the supply-following algorithm defers all jobs, while the run-immediately algorithm runs some of them. Thus, if periods of low wind are followed by periods of increased job submission, the slack of the delayed jobs may expire at the same time as new jobs that require immediate execution arrive. How often such situations occur depends on the particular mix of cluster workloads and wind power behavior, making it vitally important to use realistic wind traces and cluster workloads to quantify tradeoffs between renewable integration and performance metrics such as deadline violations.

Neither of these algorithms guarantees optimal job scheduling, i.e., always yield the highest possible percentage of wind energy to total energy used. Optimal job scheduling is impractical because it requires advance knowledge of cluster workload and wind availability, even though accurate, long-term workload predictors and wind forecasts remain elusive. Even if we have a workload and wind oracle, it is computationally infeasible to search for an optimal schedule out of all all possible job execution orders. Thus, the heuristic in the supply-following algorithm represents a compromise between optimality and practicality.



(a) Percentage of wind power usage (b) Percentage improvement (c) Percentage of delayed jobs

Figure 3: Evaluation of supply-following job scheduling

6 Evaluation

The scaling of the wind resource plays a crucial role in performance. Our raw wind traces vary between 0 and 30 MW, compared with our maximum cluster power consumption of 57.6 kW. A poor scaling factor would give trivial results. For example, if available wind power is orders of magnitude larger than what is needed by the cluster, under any scheduling algorithm 100% of energy used comes from wind. Conversely, if available wind power is orders of magnitude smaller, any scheduling algorithm would result in nearly 0% of energy coming from wind. We considered a range of scaling factors, such that the total available wind energy ranges from 0.1 to 10 times the total energy required by the cluster over the month long trace.

Figure 3(a) shows changes in the fraction of energy use that comes from wind for the two scheduling algorithms and a measure of the maximum usable wind energy given the fixed size of our cluster. Using the Pacheco wind trace, we scale the wind energy from 0.01 to 10 times the cluster's energy needs. The supply-following scheduler significantly out performs the run-immediately algorithm for all scale factors. The more wind available, the larger the performance gap. The supply-following algorithm undergoes a phase change around a wind scaling factor of 1 and exhibits diminishing returns for larger scale factors. This is likely due to the fact that as wind energy is scaled up, less of it can be used by a fixed size cluster – power spikes exceed the maximum cluster power.

Figure 3(b) shows the improvement of the supply-following versus the run-immediately algorithm for different wind traces. We compute improvement as:

$$\frac{\%\text{ energy from wind for supply-following algorithm} - \%\text{ energy from wind for run-immediately algorithm}}{\%\text{ energy from wind for run-immediately algorithm}}$$

We observe a range of improvements. At scaling factors of 1 and above, the supply-following scheduling yields a roughly 40-60% improvement.

> Key observation: The degree of renewable energy integration depends on renewable source variability and intermittence, as well as scheduling slack in the data analytic workload. Our supply-following scheduler attains a 40-60% improvement for realistic wind power and workload profiles.

To quantify how frequently the supply-following scheduling algorithm may cause jobs to exceed their deadlines, Figure 3(c) shows the percentage of all jobs that exceeded their deadlines, quantified at different wind scaling factors for the Pacheco wind trace. The percentage is very low and decreasing as the wind scaling factor increases. Also, job deadlines are never exceeded by more than one time interval, i.e. 10 minutes in our simulations. Compared with the 100s of minutes of execution time and slack shown in Figures 1(a) and 1(b), 10 minutes represents a very small amount. Thus, even though we can easily construct pathological wind traces and cluster workloads that lead to unacceptable deadline violation, for realistic wind traces and cluster workloads, deadline violations occur infrequently and have small impact.

7 Call to Arms

We must address the problem of integrating intermittent and variable renewable energy sources into the electric grid to have any hope of meeting legislative targets for renewable penetration. Current technologies and economic limits make it unlikely that we can construct load-following renewable supplies using large scale energy storage and peaker plants. We advocate for the alternative approach of constructing supply-following loads and we argue that server clusters are good candidates for tracking supplies. We have shown that simple, supply-aware scheduling algorithms can drastically increase the fraction of renewable energy consumed by data analytics clusters.

Future work includes exploring whether additional information regarding cluster workloads and wind traces can significantly improve the performance of the schedulers described in this paper. Ideally, we would like to construct a scheduling algorithm that is provably optimal and show how close to this bound practical schedulers can perform. Additionally, we want to extend our scheduler to support non-interruptible jobs and jobs with a minimum running time.

Looking forward, many opportunities and unanswered questions remain. We invite researchers and industry collaborators to implement the infrastructure for extensively tracing both cluster workloads and wind power profiles, and making such traces available. As we have shown in this paper, the level of renewable integration is highly dependent on workload and wind characteristics. Thus, having access to more cluster workloads is crucial.

Other open problems include supporting traditional DBMS or data-warehouse systems which would potentially require a different architecture. It remains an open and challenging problem to achieve power proportionality on systems that co-locate compute and storage on the same servers. We also want to consider tradeoffs between distributed supply-aware decisions made at each load, versus centralized decisions made by the electric grid operator. In this study we have assumed a data center with local, directly attached wind sources, independent of other loads. A more general scenario would consider a set of such loads.

We believe creating the information-centric energy infrastructure represents an interdisciplinary, society-wide enterprise. Computer scientists and engineers have much to contribute because of the exponentially growing energy foot-print of the technology industry, and our expertise in design, construction, and integration of large scale communication systems. Our paper demonstrates that another reason to contribute comes from the unique properties of electric loads caused by large scale computations. Consequently, data engineers in particular may end up leading the efforts to integrate renewable energy into the electric grid. We hope this paper serves as a first step in addressing this important challenge, and we invite our colleagues to join us in exploring the broader problem space.

Acknowledgements

The authors acknowledge the support of the Multiscale Systems Center, one of six research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation program. This work was supported in part by NSF Grant #CPS-0932209 and EIA-0303575, the FCRP MuSyC Center, the German Academic Exchange Service, and Amazon, eBay, Fujitsu, Intel, Nokia, Samsung, and Vestas Corporations.

References

[1] Apache hadoop. `hadoop.apache.org`.

[2] Y. Agarwal, S. Hodges, R. Chandra, J. Scott, P. Bahl, and R. Gupta. Somniloquy: Augmenting network interfaces to reduce pc energy usage. In *NSDI'09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pages 365–380, Berkeley, CA, USA, 2009. USENIX Association.

[3] L. A. Barroso. The price of performance. *ACM Queue*, 3(7):48–53, 2005.

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

[5] A. Brooks, E. Lu, D. Reicher, C. Spirakis, and B. Weihl. Demand dispatch: Using real-time control of demand to help balance generation and load. *IEEE Power and Energy Soc.*, 8(3):20–29, 2010.

[6] California Public Utilities Commission. California Renewables Portfolio Standard. `http://www.cpuc.ca.gov/PUC/energy/Renewables/`, 2006.

[7] S. Dawson-Haggerty, A. Krioukov, and D. E. Culler. Power optimization - a reality check. Technical Report UCB/EECS-2009-140, EECS Department, University of California, Berkeley, Oct 2009.

[8] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The cost of a cloud: research problems in data center networks. *SIGCOMM Comput. Commun. Rev.*, 39(1):68–73, 2009.

[9] S. V. L. Group. Data center energy forecast. `http://svlg.org/campaigns/datacenter/docs/DCEFR_report.pdf`, 2009.

[10] R. Katz, D. 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*, 2011.

[11] B. Kirby. Frequency regulation basics and trends. Technical report, Oak Ridge National Laboratory, December 2004. Published for the Department of Energy. Available via `http://www.osti.gov/bridge`.

[12] A. Krioukov, P. Mohan, S. Alspaugh, L. Keys, D. Culler, and R. H. Katz. Napsac: design and implementation of a power-proportional web cluster. In *Green Networking '10: Proceedings of the first ACM SIGCOMM workshop on Green networking*, pages 15–22, New York, NY, USA, 2010. ACM.

[13] M. Lammie, D. Thain, and P. Brenner. Scheduling Grid Workloads on Multicore Clusters to Minimize Energy and Maximize Performance. In *IEEE Grid Computing*, 2009.

[14] D. Meisner, B. T. Gold, and T. F. Wenisch. Powernap: eliminating server idle power. In *ASPLOS '09*, 2009.

[15] R. Nathuji and K. Schwan. Vpm tokens: virtual machine-aware power budgeting in datacenters. In *HPDC '08: Proceedings of the 17th international symposium on High performance distributed computing*, pages 119–128, New York, NY, USA, 2008. ACM.

[16] National Renewable Energy Laboratory. National wind technology center data. 2010.

[17] S. Nedevschi, J. Chandrashekar, J. Liu, B. Nordman, S. Ratnasamy, and N. Taft. Skilled in the art of being idle: reducing energy waste in networked systems. In *NSDI'09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pages 381–394, Berkeley, CA, USA, 2009. USENIX Association.

[18] Office of the Governor, California. Executive Order S-14-08, 2008.

[19] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for internet-scale systems. In *SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 123–134, New York, NY, USA, 2009. ACM.

[20] N. Rasmussen. Electrical efficiency modeling of data centers. Technical Report White Paper #113, APC, 2006.

[21] J. A. Roberson, C. A. Webber, M. McWhinney, R. E. Brown, M. J. Pinckard, and J. F. Busch. After-hours power status of office equipment and energy use of miscellaneous plug-load equipment. Technical Report LBNL-53729-Revised, Lawrence Berkeley National Laboratory, Berkeley, California, May 2004.

[22] R. K. Sharma, C. E. Bash, C. D. Patel, R. J. Friedrich, and J. S. Chase. Balance of power: Dynamic thermal management for internet data centers. *IEEE Internet Computing*, 9(1):42–49, 2005.

[23] G. Staples. Torque resource manager. In *SC '06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, page 8, New York, NY, USA, 2006. ACM.

[24] The Green Grid. The green grid data center power efficiency metrics: PUE and DCiE. *Technical Committee White Paper*, 2007.

[25] N. Tolia, Z. Wang, M. Marwah, C. Bash, P. Ranganathan, and X. Zhu. Delivering energy proportionality with non energy-proportional systems – optimizing the ensemble. In *Proceedings of the 1st Workshop on Power Aware Computing and Systems (HotPower '08)*, San Diego, CA, Dec. 2008.

[26] United States Senate, One Hundred Eleventh Congress. *First Session to Receive Testimony on a Majority Staff Draft for a Renewable Electricity Standard Proposal, Hearing before the Committee on Energy and Natural Resources*. U.S. Government Printing Office, February 2009.

[27] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner. Theoretical and practical limits of dynamic voltage scaling. In *DAC '04: Proceedings of the 41st annual Design Automation Conference*, pages 868–873, New York, NY, USA, 2004. ACM.