GreenMap: MapReduce with Ultra High Efficiency Power Delivery

Du Su, Yi Lu
University of Illinois at Urbana-Champaign

Abstract

Energy consumption has become a significant fraction of the total cost of ownership of data centers. While much work has focused on improving power efficiency per unit of computation, little attention has been paid to power delivery, which currently wastes 10-20% of total energy consumption even before any computation takes place. A new power delivery architecture using series-stacked servers has recently been proposed in the power community. However, the reduction in power loss depends on the difference in power consumption of the series-stacked servers: The more balanced the computation loads, the more reduction in power conversion loss.

In this preliminary work, we implemented GreenMap, a modified MapReduce framework that assigns tasks in synchronization, and computed the conversion loss based on the measured current profile. At all loads, GreenMap achieves 81x-138x reduction in power conversion loss from the commercial-grade high voltage converter used by data centers, which is equivalent to 15% reduction in total energy consumption. The average response time of GreenMap suffers no degradation when load reaches 0.6 and above, but at loads below 0.6, the response time suffers a 26-42% increase due to task synchronization. For the low-load region, we describe the use of GreenMap with dynamic scaling to achieve a favorable tradeoff between response time and power efficiency.

1 Introduction

As our reliance on online services continues to grow, so have the sizes of data centers hosting these services. As a result, energy consumption has become a significant fraction of the total cost of ownership (TCO). Electricity bills for large data centers are close to one million dollars per month in 2009 [9], and the data center energy usage in 2013 is estimated to be 91 billion kwh [16]. Consequently, data centers today contribute 2-3% of the global carbon emissions [10], and the design of environmentally friendly green data centers is an important societal need [1].

Much work on greening data centers has focused on improving computational power efficiency, where the designers strive to minimize the energy required for each unit of computation [7, 11]. For instance, energy consumption is reduced by consolidating demand onto a small number of servers [12, 8, 13] via request redirection or virtual machine migration, or by speed gating each server to optimize individual power usage [4].

On the other hand, today's data centers still use a power delivery architecture that is based on the design developed for single server applications. This conventional power delivery technique requires a very large step-down from the grid AC voltage, typically 600 or 480V AC [15], to the final CPU load of 12V DC. With today's power delivery architectures, the high voltage conversion efficiency is limited to $80-90\%$ [3, 15]. That is, $10-20\%$ of total energy consumption is wasted before any computation takes place.

Recently, a new power delivery architecture has been proposed in the power community [5]. Servers are connected in series to avoid the large step-down from the grid AC voltage, and differential power converters are used to regulate the voltage across each server. However, the differential converters incur a power conversion loss when the computational loads are *unbalanced*. The amount of loss is proportional to the difference in server computational loads. It was demonstrated in [5] that with all servers running the Linux "stress" utility, hence an almost perfectly balanced load, 99.89% power efficiency is achieved. No realistic data center traffic has been demonstrated with the new power delivery, and data center traffic is expected to display much more variation than the Linux "stress" utility.

In this paper, we explore the feasibility of the new power delivery architecture for data centers. We measured the current profile of MapReduce traffic, and observed the tremendous imbalance of computational loads

across servers. The imbalance is mainly due to the different levels of resource occupancy, and tasks at different stages consuming different amount of power.

We implemented GreenMap, a modified MapReduce framework that assigns tasks in synchronization. The preliminary work only includes results on synchronizing map tasks. The conversion losses are computed based on the measured current profile of each server.

We evaluated GreenMap with the SWIM benchmark [6], and found that at all loads, GreenMap achieves 81x-138x reduction in conversion loss from the commercial-grade high voltage converter used by data centers, which is equivalent to 15% reduction in total energy consumption. The amount of reduction from the best available high voltage converter is 27x-46x, but the best available converters are much more costly.

As GreenMap delays tasks until they can be assigned in synchronization, the average response time below 0.6 load increases by $26-42\%$. However, as load reaches 0.6 and above, no degradation in response time is observed. For the low-load region, we also describe the use of GreenMap together with dynamic scaling of data center clusters so that the load is kept around 0.6. This offers a favorable tradeoff between response time and power efficiency, while at the same time saving the energy consumption and conversion loss of idle servers.

2 Background

In data centers, the utility power has to go through several power conversion and storage elements before it reaches the servers. In a conventional power delivery architecture, the grid voltage of 600V or 480V AC is stepped down to 208V or 120V AC for distribution to racks, followed by a further conversion to DC. A DC-DC converter is installed on each server to process the full server power and to convert the high rectified voltage, typically at 208V or 120V, to a lower voltage for servers, typically 12V DC.

The large voltage step down and the need to process the full server power result in limited system-level efficiency and large converter size. The typical efficiency of a high-voltage converter used in data centers is $80-90\%$, so the conversion loss will be $10-20\%$ of the total energy consumption. The peak efficiency of the best available high-voltage converters is 95%, but they are much more costly and even larger in size [5].

Let L_{conv} denote the conversion loss in conventional converters, P the total power consumption, E the converter efficiency, V the server voltage, I_i the current in server i, and n the total number of servers, we have

$$L_{\text{conv}} = (1-E)P = (1-E)V\sum_{i=1}^{n} I_i \qquad (1)$$

where $V = 12V$,

$$E = \begin{cases} 0.8-0.9 & \text{for converters in data centers,} \\ 0.95 & \text{for best available converters.} \end{cases}$$

Recently, a new power delivery architecture has been proposed in [5]. Instead of employing a high-voltage step-down for each server, a set of n servers are connected in series to equally share the rectified grid voltage. For a suitable choice of n, the series-stacked architecture provides an inherent step-down, where each server's input voltage is $1/n$ fraction of the grid voltage.

However, as the series-connected servers conduct the same current, this leads to a variation in server voltage even if there is only a small mismatch in power consumption between servers. Regulated voltage of all servers in the series stack can be achieved through the use of differential power converters, one for each of the servers. Instead of processing the full server power, the differential power converters only process the difference between the power of each server and the average power in the series stack. As a result, the efficiency of the power converter can be made as high as the best available converter, that is, 95%, at a reasonable cost, and the differential power converters are of a much smaller size.

Let L_{diff} denote the conversion loss in differential converters. The server voltage V can be considered constant at 12V due to voltage regulation, we have

$$\begin{aligned} L_{\text{diff}} &= 1.5(1-E)V\sum_{i=1}^{n}(I_i - I_{\text{avg}}) \qquad (2) \\ \text{where} \quad & V = 12V,\ E = 0.95, \\ & I_{\text{avg}} = \frac{1}{n}\sum_{i=1}^{n} I_i. \end{aligned}$$

The extra factor of 1.5 is due to the specific topology of the server-to-virtual-bus differential power converter [5], where the secondary side of the differential converter is connected to a virtual bus. As shown by the term $(I_i - I_{\text{avg}})$, the more balanced the computational loads are (hence the server currents), the smaller the power conversion loss will be.

3 Load Balancing

The series stack can be integrated into data center racks as illustrated in Figure 1. As servers are added to data center in racks, a rack can consist of more than one series stacks. This facilitates the installation of a series-connected stack and provides proper ground isolation [5]. The server hosting the resource manager (RM) is not in a series stack, as its computational load is very different from the other servers.

The number of servers in a series-stack is upper bounded by the ratio of rectified grid voltage to the server voltage. Apart from the rectified grid voltage of 600V or 480V, we can also use intermediate DC voltages utilized in many data center implementations. For instance,

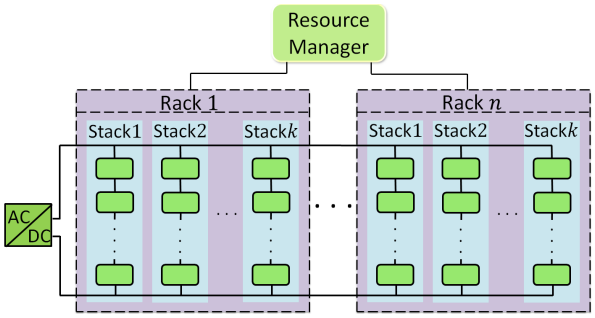


Figure 1: Data center with series-connected stacks.

48V is a standard telecom supply voltage. In this paper, we will compute power conversion loss based on a series stack of 4 servers, as this is the experimental setup in [5]. This corresponds to a voltage of 48V across the series-stack.

3.1 Current Profiling

We start by profiling the power consumption of a word-count job containing one map task and one reduce task on a server with conventional power supply. The server voltage is fixed to 12V. We measure the current consumption of the server using a Yokogawa wt310 digital power meter. Figure 2 shows the current consumption at different stages of a word-count job.

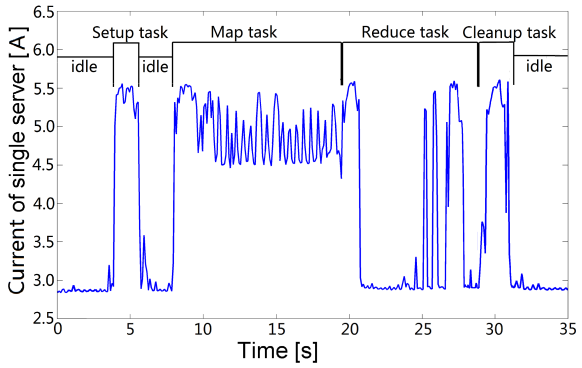


Figure 2: Current consumption of a word-count job with one map task and one reduce task.

The idle current is around 2.8A. The setup task initializes the job and creates temporary output directories, consuming close-to-peak current at 5.5A for 2.5s. The server goes idle for another 2.5s before launching the map task. The beginning of the map task consumes close-to-peak current as a new thread is initialized and data are read into memory. However, the bulk of the map task experiences an oscillation of current around 4.8A, as it generates <key,value> pairs and outputs them to the intermediate directory. The alternating computation-intensive and I/O-intensive operations cause the current to oscillate. The beginning of the reduce task also consumes close-to-peak current as a new thread is initialized, followed by 4 seconds of low current at 2.8A, as <key,value> pairs are copied from intermediate directories on other servers. The later stage of the reduce task is characterized by large oscillations between 2.8A and 5.5A as the high-current computation-intensive operations intersperse among the low-current I/O operations. The cleanup task after the job's completion causes another short period of close-to-peak current consumption.

In general, a MapReduce job always has a setup task and a cleanup task. It can have multiple map tasks and reduce tasks, whose current consumption can vary depending on user-defined functions, although map tasks (or reduce tasks) of the same job will still have similar current profiles.

3.2 Synchronized Task Assignment

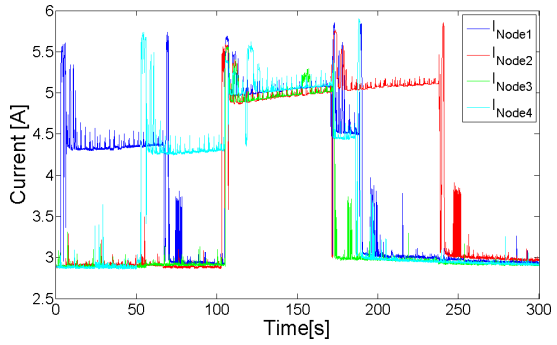
We built GreenMap to balance the computational loads in a series-stack by synchronizing map task assignment. There are three main modifications to the default MapReduce scheduler.

First, the setup and cleanup tasks are moved to the server where the RM resides. As each setup task (and cleanup task) is executed only once per job, and it consumes close-to-peak power, it is inherently unsuitable for parallelization and balancing across a series-stack of servers. Although we co-locate setup and cleanup tasks with the RM in this experiment, in a more scalable implementation, they can be assigned to any server outside series-stacks. A data center can consist of series-stacks on which parallelized tasks run, and conventional servers for tasks unsuitable for parallelization.

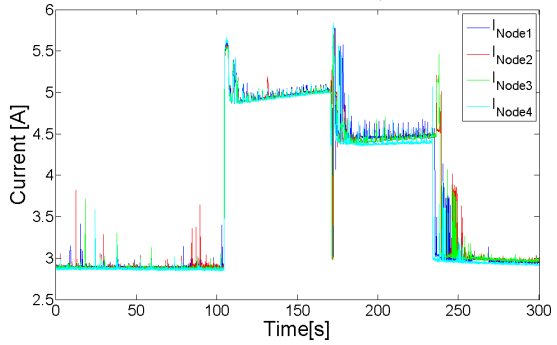
Second, we minimize load imbalance by assigning the same number of map tasks to each server, and whenever possible, assigning map tasks of the same job in synchronization. This is achieved by delaying task assignment until the number of outstanding tasks is at least that of the servers with idle slots. In particular, when there exist outstanding jobs, a server with an idle slot will be assigned a task immediately, in accordance with the assignment by the default FIFO scheduler based on data locality. When there exist no outstanding jobs, the number of servers with idle slots increases over time. At a new job arrival, if the number of outstanding tasks exceeds that of idle servers, a batch of tasks are assigned in synchronization. If the tasks are insufficient to fill all idle servers, they are delayed till further jobs arrive.

Third, to prevent the system from delaying tasks for too long, a timer is set to zero whenever tasks are assigned in synchronization or a new job has arrived. In the absence of neither, when the timer reaches a threshold value, all outstanding tasks are assigned. The threshold is set to be the time period during which a new job will arrive with 90% probability at the current load. The

exact value of the threshold is not important and has not been optimized for this experiment. A larger value for the threshold will further reduce power conversion loss and increase response time, while a smaller value will increase power loss and reduce response time.



(a) Imbalanced loads with no task synchronization



(b) Balanced loads with task synchronization

Figure 3: Current profiles of four servers.

To demonstrate the effect of task synchronization, we ran a small trace on four servers with conventional power delivery, and compared the measured current profiles. The trace consists of two jobs of 1 map task, one job of 2 map tasks and one job of 8 map tasks, arriving at random intervals. Figure 3(a) shows the measured current profiles of the four servers respectively, with no task synchronization. Not surprisingly, we observe large difference in currents consumed at each server. Figure 3(b) shows the current profiles of the same four servers with synchronized task assignment. We observe that the map tasks are indeed synchronized, and the difference in currents consumed at different servers becomes small and only occurs at sporadic moments.

4 Evaluation

Our test bed includes five Dell Optiplex SX775 Core 2 Duo workstations. One server hosts the Resource Manager (RM) and is not in a series-stack. The remaining four servers simulate a series-stack of 48V.

GreenMap is implemented in Hadoop1 for this preliminary work as the centralized design of Hadoop1 scheduler is amenable to task synchronization. Each server has 2 map slots. We do not consider reduce tasks in this experiment.

We generate traces by selecting jobs from the SWIM benchmark [6] so that we achieve a good representation of the Pareto job size distribution [2], and the length of the trace and the number of files are appropriately scaled for the capacity of one series-stack. Job arrivals are generated as a Poisson process, and each job does not contain any reduce tasks. The data block size is set to 32 MB, and each map task takes an average of 70 seconds. Hence for each load point, the trace takes $1.5-6$ hours on our cluster. After scaling, our trace contains 447 tasks and 50 jobs. Table 1 shows the job size distribution.

Bins	1	2	3	4	5	6
Job count	25	9	6	4	3	3
Map count per job	1	2	4	8	16	100

Table 1: Job size distribution.

We connect the 4 servers with a conventional power delivery architecture, and measure the current consumption of each server using a Yokogawa wt310 digital power meter with 10 samples per second per server. We compute the power conversion loss using equations (1) and (2). The advantage of this setup is that it allows us to compare the conventional conversion loss and the differential conversion loss in the exact same setting, with the same run of a trace. Note that the conventional conversion loss depends on the sum of the current, whereas the differential conversion loss depends on the deviation of each current from the average current in the stack.

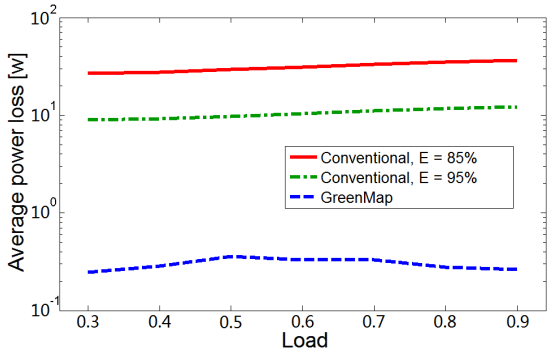


Figure 4: GreenMap reduces power conversion loss from the conventional architecture by two orders of magnitude.

Figure 4 shows that at all loads, GreenMap achieves 81x-138x reduction in conversion loss from the conventional power delivery with a commercial-grade high voltage converter of 85% efficiency, which is typical of converters used in data centers today. The power conversion loss is reduced by two orders of magnitude, from an average of 31.4W to 0.3W. This is equivalent to 14.999% reduction in total energy consumption, almost eliminating the 15% conversion loss altogether.

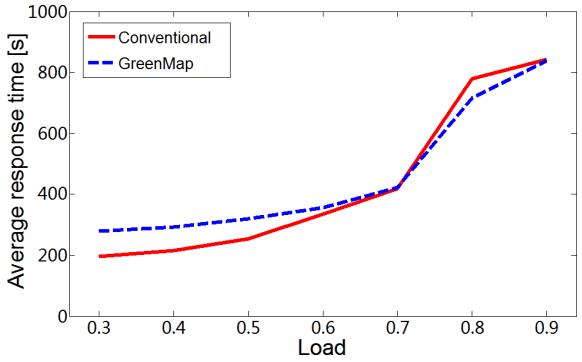


Figure 5: GreenMap achieves comparable response time as Hadoop FIFO scheduler at load 0.6 and above, while increasing response time at lower loads.

Figure 4 also shows the conversion loss of the conventional power delivery with the best available high-voltage converter of 95% efficiency. GreenMap achieves 27x-46x reduction in power conversion loss, from an average of 10.45W to 0.3W.

Figure 5 shows the average job response time of the default Hadoop FIFO scheduler versus that of GreenMap. As GreenMap delays task assignment until tasks can be assigned in synchronization, the average response time below 0.6 load increases by $26-42\%$. However, when the load reaches 0.6 and above, no degradation in response time is observed. This is because there are an abundance of outstanding tasks at high loads, and tasks are seldom delayed, whereas the sparse arrivals of tasks at low loads result in more tasks being delayed.

4.1 GreenMap with Dynamic Scaling

The above results show that GreenMap suffers a degradation in response time when the load is below 0.6. In fact, GreenMap delays tasks in order to emulate a higher load, at which there are an abundance of outstanding tasks, hence facilitating assignment in synchronization. We observe that higher loads can be more efficiently achieved by turning off a fraction of stacks in a large cluster with multiple series-stacks.

For instance, assume that we have 10 series-stacks running at 0.4 load. From Figure 4, the total power consumption in each series-stack of 4 servers is 192.2W ($=V\sum_{i=1}^{4} I_i$) at 0.4 load, and the conversion loss in each series-stack is 0.29W with GreenMap. With dynamic scaling, we can turn off 3 series-stacks, resulting in 0.57 load for each of the remaining series-stack. The corresponding power consumption in each series-stack is now 215.4W, and the conversion loss is 0.33W with GreenMap. Hence, with GreenMap but not dynamic scaling,

$$\text{total power } = (192.2+0.29)\times 10 = 1924.9\text{W},$$

whereas with GreenMap and dynamic scaling,

$$\text{total power } = (215.4+0.33)\times 6 = 1294.4\text{W},$$

which is a 32.8% reduction. The reduction in total energy consumption is similar as the servers are mostly idle at 0.4 and 0.57 load, and the trace takes a similar amount of time to finish. The average job response time with GreenMap will increase by only 15% as the load increases from 0.4 to 0.57, yielding a favorable tradeoff between power efficiency and response time.

5 Related Work

A new DC-DC power delivery architecture was proposed in [15]. Instead of the conventional step-down of 600V or 480V AC to 208V or 120V AC for distribution to racks, followed by a further conversion to DC for energy storage, the DC-DC architecture uses a single rectification stage. It boosts the efficiency of the best-in-class AC-DC power supply from 90% to 92%. However, the power conversion loss is still directly proportional to the total energy consumption in the entire data center, while GreenMap's conversion loss only depends on the difference in power consumption, which makes it possible to achieve an ultra-high efficiency of 99%.

Another power delivery architecture also using differential power converters is proposed in [14]. Instead of connecting each server in a series-stack to a virtual bus, this architecture connects neighboring pairs of servers in the series-stack. While this architecture achieves comparable conversion efficiency as the architecture proposed in [5], the conversion loss depends on the difference in power consumption of neighboring servers, instead of the difference between each server and the average. This makes the load balancing problem much more difficult as finding the optimal assignment becomes a combinatorial problem involving the location of a server in the series-stack.

6 Conclusion

We explored the feasibility of series-connected stacks in data centers by implementing GreenMap, a modified MapReduce framework that assigns tasks in synchronization. We found that with task synchronization, the conversion loss in data centers can potentially be reduced by two orders of magnitude, which is equivalent to about 15% of total energy consumption. Future work includes implementing GreenMap with multiple series-stacks and heterogeneous jobs, and evaluating the system on actual series-connected stacks.

Acknowledgement

We thank Enver Candan for his assistance with setting up the Yokogawa wt310 digital power meter. Yi Lu is supported by NSF grant CNS-1150080.

Discussion Topics

The most important feedback we seek to receive from this community is the relative priority of energy saving, performance and ease of implementation of future scheduling software.

For instance, how much performance are we willing to sacrifice for 15% reduction of total energy consumption? The sacrifice is in relative terms rather than in absolute terms: The same logic applies to utilization. While appropriate packing algorithms can increase utilization without degrading performance, beyond a certain threshold, utilization of a cluster is tied with response times due to the stochastic nature of job arrivals. Do we want to run an almost empty cluster at 0.2 load and achieve the best possible response time? Or do we rather run the cluster at 0.7 load with a 10% increase in response time? What if we have a clever scheduling algorithm that will reduce the response time at all loads? With a clever algorithm, the absolute sacrifice might disappear. However, the relative sacrifice always exists because the performance of the clever algorithm will always be better at 0.2 load than at 0.7 load.

With this in mind, it might be even more important to consider whether the 15% energy saving is worth the additional constraint imposed on the design of scheduling algorithms. GreenMap will require computational loads to be balanced in each series-stack in order to minimize power conversion loss. For instance, it might not be easy to implement the YARN architecture on series-stacks, as map and reduce tasks co-locate with application managers (AM), whose loads might be very different from one another.

Another related question will be how many servers a series-stack should contain, as it determines the granularity at which batches of tasks are assigned. In an environment with heterogeneous applications, a small granularity will lessen the constraint on scheduling algorithms, while potentially introducing a higher power loss. A larger granularity, on the other hand, will make it more difficult to balance computational loads on all servers in the series-stack.

References

[1] Report to congress on server and data center energy efficiency opportunities. *http://www.energystar.gov/* (2007).

[2] ABAD, C. L., ROBERTS, N., LEE, K., LU, Y., AND CAMPBELL, R. H. A storage-centric analysis of mapreduce workloads: File popularity, temporal locality and arrival patterns. In *IEEE International Symposium on Workload Characterization (IISWC)* (2012).

[3] ALDRIDGE, T., PRATT, A., KUMAR, P., DUPY, D., AND G., A. Evaluating 400v direct-current for data centers – a case study comparing 400 vdc with 480-208 vac power distribution for energy efficiency and other benefits. Tech. rep., Intel Labs.

[4] ANDREW, L. L. H., LIN, M., AND WIERMAN, A. Optimality, fairness, and robustness in speed scaling designs. In *Proc. of Sigmetrics* (2010).

[5] CANDAN, E., SHENOY, P. S., AND PILAWA-PODGURSKI, R. C. A series-stacked power delivery architecture with isolated differential power conversion for data centers. In *Telecommunications Energy Conference (INTELEC), 2014 IEEE 36th International* (2014), IEEE, pp. 1–8.

[6] CHEN, Y., GANAPATHI, A., GRIFFITH, R., AND KATZ, R. The case for evaluating MapReduce performance using workload suites. In *Proc. IEEE Int'l Symp. Modeling, Analysis & Simulation of Comput. Telecomm. Syst. (MASCOTS)* (2011).

[7] FAN, X., WEBER, W.-D., AND BARROSO, L. A. Power provisioning for a warehouse-sized computer. In *ISCA'07* (2007).

[8] GANDHI, A., HARCHOL-BALTER, M., RAGHUNATHAN, R., AND KOZUCH, M. Distributed, robust autoscaling policies for power management in compute intensive server farms. In *Open Cirrus Summit 2011* (2011).

[9] GREENBERG, A., HAMILTON, J., MALTZ, D., AND PATEL, P. The cost of a cloud: Research problems in data center networks. *ACM Sigcomm Computer Communication Review (CCR)* (2009).

[10] Smart 2020: Enabling the low carbon economy in the digital age.

[11] HOELZLE, U., AND BARROSO, L. A. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 1st ed. Morgan and Claypool Publishers, 2009.

[12] KRIOUKOV, A., MOHAN, P., ALSPAUGH, S., KEYS, L., CULLER, D., AND KATZ, R. Napsac:design and implementation of a power-proportional web cluster. In *Proc. of Sigcomm workshop on Green Networking* (2010).

[13] LIN, M., WIERMAN, A., ANDREW, L. L., AND THERESKA, E. Dynamic right-sizing for power-proportional data centers. *Proc. IEEE INFOCOM, Shanghai, China* (2011), 10–15.

[14] MCCLURG, J., PILAWA-PODGURSKI, R. C., AND SHENOY, P. S. A series-stacked architecture for high-efficiency data center power delivery. In *Energy Conversion Congress and Exposition (ECCE), 2014 IEEE* (2014), IEEE, pp. 170–177.

[15] TON, M., FORTENBERY, B., AND TSCHUDI, W. Dc power for improved data center efficiency. Tech. rep., Lawrence Berkeley National Laboratory, 2008.

[16] WHITNEY, J., AND DELFORGE, P. Data center efficiency assessment. Tech. rep., Natural Resources Defense Council, Aug. 2014.