Policy and Mechanism for Carbon-Aware Cloud Applications

Nan Deng and Christopher Stewart
The Ohio State University

Daniel Gmach and Martin Arlitt
Hewlett Packard Labs

***Abstract*—This position paper explores research challenges facing carbon-aware cloud applications. These applications run inside of a renewable-energy datacenter, provision resources on demand, and seek to minimize their use of carbon-heavy, grid energy. First, we argue that carbon-aware applications need new provisioning policies that address the uncertainty of renewable energy. Second, we argue that renewable-energy datacenters need mechanisms to determine the contribution of grid energy to specific application workloads. We propose first-cut solutions to these problems and present preliminary results for carbon-aware Web server running on a small renewable-energy cluster.**

I. Introduction

A growing cadre of green datacenters now use rooftop solar panels and on-site wind turbines [11], [8], to reduce their carbon footprint. Since these datacenters can not rely solely on intermittent renewable energy, they often tie into the electric grid to achieve high availability and to ensure that every renewable joule produced can power datacenter servers [19]. Increasingly, individual applications within the datacenter are under pressure to reduce their own carbon footprints by using such renewable-powered servers. For example, over 500,000 Facebook members have signed a petition pressuring the site's managers to use only renewable energy [20]. Similarly, Greenpeace.org has pressured Twitter by giving them an F (failing) grade in an industry-wide study of carbon mitigation [6]. We believe that these dismal results reflect an unmet technical challenge, not a disregard for the environment.

This position paper outlines our research on carbon-aware policy and mechanism. We target applications that provision datacenter resources on demand, e.g., on cloud platforms like EC2 [1]. We call these *cloud applications*. Policies are the rules governing provisioning decisions for an application. Mechanisms, provided by the datacenter, define and control the provisioning process.

Our proposed carbon-aware cloud applications treat carbon-heavy energy as a primary cost, provisioning a cloud instance only if its emissions costs are justified by application-specific rules. Emissions costs can vary unpredictably over time with the production of renewable energy, adding noise to the decision-making process. We propose carbon-aware policy that can be 1) expressed in formal models and 2) applied to a wide range of practical scenarios. As formal models, the impact of uncertain emissions costs can be analyzed rigorously across weather patterns, datacenters, and application workloads. On the practical side, our policies support diverse performance and cost goals. Section III proposes a model for carbon-aware cloud applications.

In renewable-energy datacenters tied to the electric grid, compute resources can get power from both the grid *and* renewable sources at the same time. Section IV proposes a mechanism to attribute carbon-heavy grid energy to specific, provisioned resources (and hence to applications). The challenge here is that grid ties make electricity from renewable and carbon-heavy sources functionally indistinguishable. Our mechanism uses proportional sharing [4], a well-known power engineering principle, which assigns carbon-heavy energy to a datacenter resource in proportion to the net contribution of grid power flowing into the resource. We implement this mechanism operationally, using direct measurements of the flow of grid energy with and without on-site sources. As a proof of concept, we've profiled a small experimental renewable-energy cluster and studied carbon-aware policy for the Apache Web server under the 1998 World Cup workload [3].

II. Related Work

Several recent research papers have explored carbon awareness in cloud datacenters [19], [17], [14]. In these prior works, cloud resources received all of their power from either renewable energy or carbon-heavy grid energy. On-site renewable sources were isolated from the electric grid. Sharma et al. [17] simply turned off cloud resources when there was not enough renewable energy to power them completely. Li et al. [14] adjusted the energy demand (e.g., DVFS) of microprocessors, hoping to power them under low renewable-energy production. Gmach et al. [10], [9] used server-power capping and consolidation to power servers under low renewable-energy production. Stewart et al. [19] described the limits of such transfer-switched approaches, showing that renewable-energy production below a resource's threshold is either wasted [14] or must be stored in costly batteries [17], [10].

Our work examines carbon-aware applications in grid-tied datacenters. In these datacenters, cloud resources can use electricity from the grid and on-site sources, simultaneously. Grid ties present two core research challenges. First, carbon-aware policy must consider fractional contributions from on-site sources. Second, datacenters must devise fair carbon accounting mechanisms that attribute carbon-heavy grid energy to specific applications.

Other related work has examined carbon-aware grid-tied datacenters [13], [15]. Looking at the carbon footprint of entire datacenters, these works measure the contribution of renewable energy from utility providers over time. By dynamically migrating workload to the datacenter with the least carbon-heavy, grid energy, the aggregate footprint for a collection of datacenters is reduced. Compared to our work, these whole datacenter approaches also allow grid energy to supply a fraction of total energy needs. However, the policy decisions faced by application managers (e.g., to use a resource or not) are unlike those faced by datacenter owners (e.g., which datacenter to use). Further, whole-datacenter approaches do not need to account for the carbon footprint of cloud resources, since these approaches target the datacenter's total carbon footprint.

III. Policy: Carbon-Aware Provisioning

Before introducing the carbon-aware provisioning problem, we first define a few terms for readers without a background in cloud computing. A *cloud resource* is a collection of computing hardware within a datacenter, e.g., CPU, disk, and memory. A cloud resource running for a fixed period of time is a *cloud instance* . An *application manager* provisions cloud instances that collectively compose a *cloud application* .

In carbon-aware provisioning, the application manager sets a hard limit on the total carbon-heavy, grid energy of provisioned cloud instances. Carbon-aware policy decides which cloud instances to provision such that 1) the grid energy limit is not exceeded and 2) performance goals are met. In this paper, we focus on throughput (i.e., requests per second or job processing rate) as performance metrics. Carbon-aware policy needs estimates of grid-energy needs, performance goals, and grid-energy caps. These can be adjusted at each time interval where cloud instances can be provisioned.

Mathematical Model: Suppose there are n cloud instances available for provisioning. At time period t, a provisioning strategy for an application is denoted as a vector $\mathbf{x^{(t)}} = (x_1^{(t)}, x_2^{(t)} \cdots x_n^{(t)})^T$, in which $x_i^{(t)} = 1$ if the ith instance is provisioned, otherwise $x_i^{(t)} = 0$. Each cloud instance can provide up to its *maximum throughput* for a target application, represented by the vector $\mathbf{v} = (v_1, v_2 \cdots v_n)^T$. v_i is a real number that denotes the maximum throughput provided by the server which will run ith instance. The *maximum throughput* of a cloud application is the summation of the maximum throughput of its provisioned instances, i.e. $\mathbf{v}^T\mathbf{x^{(t)}}$. Here, the symbol T is the transpose function in vector multiplication. We also note that some applications, e.g., Web servers, may have fluctuating throughput needs over time. We use $V^{(t)}$ to represent the target throughput at time t.

The hard limit on carbon-heavy, grid energy is $D^{(t)}$ as set by the application manager. The total grid energy used by an instance at time t is represented as a vector $\mathbf{d^{(t)}} = (d_1^{(t)}, d_2^{(t)} \cdots d_n^{(t)})^T$. Note, $d_i^{(t)}$ is a real number between zero and max energy needs of the ith instance. An application's total grid energy consumption at time t is $\mathbf{d}^T\mathbf{x^{(t)}} < D^{(t)}$.

Given $\mathbf{v}, \mathbf{d^{(t)}}, D^{(t)}, V^{(t)}$, our goal is to find $\mathbf{x^{(t)}}$. Specifically, we represent carbon-aware provisioning as an integer programming problem:

$$\text{Maximize } \mathbf{v}^T\mathbf{x^{(t)}} \tag{1}$$

$$\text{Subject to } \mathbf{d^{(t)}}^T\mathbf{x^{(t)}} \leq D^{(t)} \tag{2}$$

$$\text{and } \mathbf{v}^T\mathbf{x^{(t)}} \leq V^{(t)} \tag{3}$$

This problem is a variant of the well-known NP-complete knapsack problem [12]. A deterministic algorithm that finds optimal provisioning strategies quickly (i.e., in polynomial time) is unlikely. However, algorithms that quickly find approximate solutions exist [12]. Figure 1 shows a recursive dynamic programming solution [16]. Here, the function $f(i,j)$ means the maximum throughput (less than $V^{(t)}$) of the application if we only consider the first i instances. The initial value of j is the manager's limit on carbon-heavy energy.

$$f(i,j) = \begin{cases} f(i-1,j) & \text{if } d_i^{(t)} > j \\ & \text{or } d_i^{(t)} \leq j \\ & \quad \text{and } f(i-1,j) > v_i + f(i-1, j-d_i^{(t)}) \\ & \text{or } d_i^{(t)} \leq j \\ & \quad \text{and } f(i-1,j) \leq v_i + f(i-1, j-d_i^{(t)}) \\ & \quad \text{and } f(i-1, j-d_i^{(t)}) + v_i > V^{(t)} \\ & \quad \text{and } f(i-1,j) > f(i,j-1) \\ f(i,j-1) & \text{if } d_i^{(t)} \leq j \\ & \text{and } f(i-1,j) \leq v_i + f(i-1, j-d_i^{(t)}) \\ & \text{and } f(i-1, j-d_i^{(t)}) + v_i > V^{(t)} \\ & \text{and } f(i-1,j) \leq f(i,j-1) \\ f(i-1, \\ \quad j-d_i^{(t)}) + v_i & \text{otherwise} \end{cases}$$

Fig. 1: Dynamic Programming Solution

The boundary condition is $f(0,0) = 0$. The dynamic programming algorithm uses 2-level loop, iterating from $i = 0$ to $i = n$; and in each i loop, iterating j from 0 to $D^{(t)}$. The algorithmic runtime for this solution is $O(nD^{(t)})$.

In practice, each resource's dependence on carbon-heavy energy ($\mathbf{d^{(t)}}$) changes unpredictably with the weather. Inaccurate predictions will affect carbon-aware policy. Extending our earlier problem definition, let $d_i^{\hat{(t)}}$ denote the actual grid-energy usage by ith instance at time t. Let $\Delta d_i^{(t)}$ be a random variable denoting the difference between actual grid-energy usage at time t ($d_i^{\hat{(t)}}$) and the predicted grid-energy usage at time t ($d_i^{(t)}$). We use vectors $\Delta\mathbf{d^{(t)}}$ and $\mathbf{d}^{\hat{\mathbf{(t)}}}$ to represent $(\Delta d_1^{(t)}, \Delta d_2^{(t)} \cdots \Delta d_n^{(t)})^T$ and $(d_1^{\hat{(t)}}, d_2^{\hat{(t)}} \cdots d_n^{\hat{(t)}})^T$, respectively. Our new extended problem definition is:

$$\text{Maximize } \mathbf{v}^T\mathbf{x^{(t)}} \tag{4}$$

$$\text{Subject to } {\mathbf{d}^{\hat{\mathbf{(t)}}}}^T\mathbf{x^{(t)}} - \Delta{\mathbf{d^{(t)}}}^T\mathbf{x^{(t)}} \leq D^{(t)} \tag{5}$$

$$\text{and } \mathbf{v}^T\mathbf{x^{(t)}} \leq \hat{V}^{(t)} + \Delta V^{(t)} \tag{6}$$

From Equation (5), we can see that once we make the decision of our strategy, the real upper bound of the dirty energy used is $D^{(t)} + \Delta{\mathbf{d^{(t)}}}^T\mathbf{x^{(t)}}$. Hence $\Delta{\mathbf{d^{(t)}}}^T\mathbf{x^{(t)}}$ is the error we may introduce. Using this model of uncertainty as a base, we plan to answer the following research questions:

1. Which probability distribution characterizes $\Delta d_i^{(t)}$? The answer depends on the prediction method [18] and the on-site renewable-energy source (e.g., solar versus wind). However, both solar and wind energy are subject to sudden unexpected outages that lead to heavy-tail production patterns [19]. Heavy-tail production often leads to heavy-tail prediction error also.

2. Do different instances have different $\Delta d_i^{(t)}$? If on-site renewable energy is evenly distributed to all resources, we would expect the answer to be no. However, our recent work [7] shows that grid-tie placement can concentrate renewable energy to certain datacenter resources. Our results in Section IV confirm this result using a different mechanism.

If the answer is yes, $\Delta d_i^{(t)}$ may be statistically dependent between instances.

Note that our model naturally extends to the practical case where instances are heterogeneous. However, in the simple homogeneous case—if renewable energy is evenly distributed— then we can assume that $\Delta\mathbf{d^{(t)}}$ are independent and identically distributed random variables. If there are large enough number of instances running the service, it is safe to say that $\Delta\mathbf{d^{(t)}}^T\mathbf{x^{(t)}}$ is a normally distributed random variable according to central limit theory. This condition may be important for datacenters that apply renewable energy incrementally by first targeting a few homogeneous racks.

IV. Mechanism: Carbon Accounting

The carbon-aware policy discussed in Section III supports grid-tied datacenters where the electric grid contributes fractionally to energy needs. These contributions were $d_i^{(t)}$, a parameter that could range from 0 to the energy consumption of resource i. Carbon accounting is the mechanism that attributes carbon-heavy electricity to cloud resources. As we began to implement carbon-aware applications, we found that grid ties complicate carbon accounting.

Grid-Tied Datacenters: The DC electricity produced by solar panels is incompatible with the AC electricity pulled from the electric grid. Grid ties solve this problem by inverting DC electricity from on-site sources and producing AC electricity that matches the grid's voltage and frequency. Electricity from these two sources become indistinguishable to datacenter devices which draw power as they normally would. Grid ties are necessary for net metering, a widely used concept in which owners of solar panels and wind turbines can be reimbursed for contributing electricity to the grid—a potential revenue stream for datacenter owners. Also, the cost of grid ties does not increase at scale, unlike, batteries for energy storage. Finally and critically, grid ties can be deployed incrementally. Allowing datacenters to slowly invest in renewable energy. However, grid ties make it difficult to infer the exact contribution of carbon-heavy sources. Accurate and technical carbon accounting must underlie any practical carbon-aware system.

Operational Carbon Accounting: Our proposed solution measures the net energy pulled from the grid before and after grid ties are installed. Since grid ties do not affect the energy needs of datacenter devices, the amount of electricity flowing through power-delivery devices upstream relative to grid ties must decrease (according Kirchhoff's current law [5]). The proportional sharing principle [4] attributes energy from multiple sources to individual sinks (i.e., cloud resources) according to each source's relative contribution to total energy needs. In our solution, the grid contributes the net grid energy pulled *after* a grid tie is installed. The total energy needs are the net grid energy pulled *before* a grid tie is installed.

Under a net-energy model, grid-tie placement affects the concentration of grid energy used to power cloud resources [7]. A fixed injection of electricity causes greater relative reduction in grid energy when there are few devices downstream (i.e., a small denominator). A grid tie could inject more electricity on the circuit than the downstream resources need. In this case, the electricity flowing through power delivery devices further upstream decreases recursively.

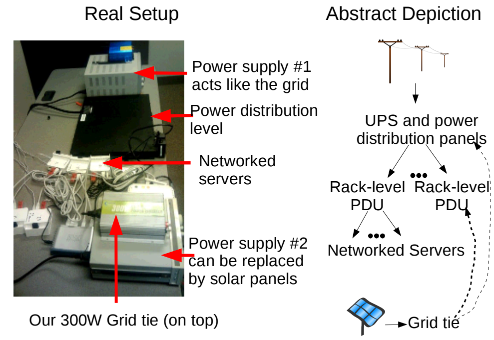

Fig. 2: Our renewable-energy cluster; the real setup and abstract goal. Dotted lines reflect grid-tie placements. Note to reviewer, we would like to host a demo at the conference.

V. Preliminary Results

Combining policy and mechanism, we have set up an experimental renewable-energy cluster in our lab (shown in Figure 2). The top component in Figure 2 is a power supply unit with capacity that easily exceeds the max power of our cluster. This power supply 1) serves as our experimental grid and 2) isolates electricity produced by our grid tie (ensuring proper, downstream-only usage of the real grid). The bottom component is a programmable power supply that connects to a grid tie; it can be replaced by a real solar panel (i.e., for demo). Figure 2 also shows how these components relate to traditional power delivery within the datacenter. It allows us to produce repeatable results. Our cluster has 5 low-power Linux servers, each with Apache and MapReduce capabilities, a 1GbE switch, and a 1GbE router. Collectively, our cluster consumes about 42W, excluding the power supply. We used clamp-on amp meters and line splitters to measure the power usage at each link.

Mechanism Experiments: A cloud resource in our renewable-energy cluster is 1 server (1.2Ghz processor, 128 MB, 1 Gb Ethernet). Power delivery devices in our renewable-energy cluster are power strips and 1 rack-style PDU. Our grid tie needs and produces 120V AC, making it compatible with our servers. This allows us to measure the effect of grid-tie placement on net grid energy. Here, we present an experiment that compares two grid-tie placements. The first places the grid tie just below our grid-like power supply, a top-level placement that evenly dilutes the concentration of energy from on-site sources [7]. The second places the grid tie on a circuit where cloud resources 2–4 are downstream, but resources 0–1 are powered in parallel on a different circuit. This changes the net contribution of renewable energy between the two groups. Figure 2 depicts grid-tie placement.

Figure 3 shows the grid energy used by nodes 1 and 2 in our experiment. We injected different amounts of "renewable" energy using our programmable power supply. The grid-tie placement changed the distribution of renewable energy under our carbon accounting approach. The low-level placement sent

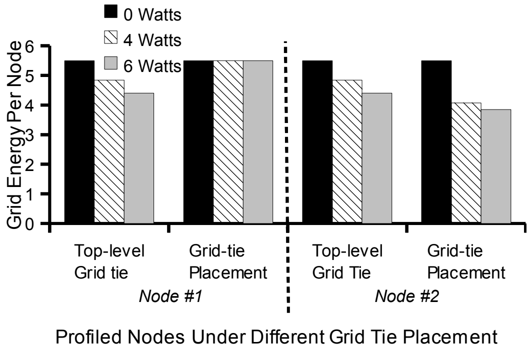

Fig. 3: Net grid energy used by nodes 1 and 2 in our cluster. The legend shows the amount of power injected by the grid-tied power source.

more renewable energy to nodes 2–4 by taking from nodes 0–1, increasing the sustainability of node 2 by 14%.

Policy Experiment: We set up the Apache Web Server [2] on our cluster. The workload matched a scaled version of the World Cup 1998 logs [3]. We provisioned instances by restarting Apache on a cluster node and telling our load balancer to direct requests to it. We unprovisioned instances by stopping Apache. The maximum throughput of each cloud resource (v_i) was measured in requests per second (rps) via offline profiling. Our homogeneous cluster nodes supported 15 rps before the average response time exceeded our limit of 100ms. We also considered a *heterogeneous* cluster where hypothetical profiling provided the same maximum throughput (75 rps) but with varied rps per node (from 12–20). Building on Figure 3, we configured our cluster for a grid-tie placement that gave more renewable energy to nodes 2–4 under our operational carbon accounting. We set the contribution of renewable energy to 10% ($D^{(t)}$ is high) and 55% ($D^{(t)}$ is low) of the cluster's maximum energy.

Figure 4 has important implications for application managers concerned about sustainability. First, the heterogeneous cluster achieves higher throughput than the homogeneous. This is because the heterogeneous cluster supports diverse combinations of instances, and our carbon-aware policy naturally controls this parameter. Second, strong (low) carbon caps degraded performance proportionally relative to the cap, whereas weak (high) carbon caps had disproportionate effects. Under a low carbon cap (45% of total power), the observed throughput under the heterogeneous cluster saturated at 43% of the maximum. However, under a high carbon cap (90% of total power), the observed throughput dropped to only 82% of the maximum. Here again, low carbon caps allowed our integer-programming approach to choose from more instance combinations. These preliminary results indicate that an application looking to reduce its carbon footprint should prefer datacenters that supply diverse instances.

VI. Conclusion

The carbon footprints of datacenters and their hosted applications are now heavily scrutinized. Carbon-aware cloud applications take a step toward environmentally friendly computing by treating carbon-heavy energy as a primary cost.

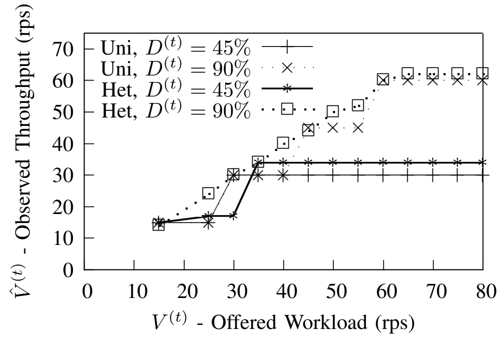

Fig. 4: Observed throughput versus offered workload for Apache under a carbon-heavy energy cap. Uni is the homogeneous cluster; Het stands for heterogeneous.

In providing support for carbon awareness, we found the classic systems principle of separating policy and mechanism useful. Specifically, new datacenter mechanisms are needed to distribute dirty grid-energy to individual applications. We provide an approach that is operational and produces reasonable, intuitive results. Also, we explore integer programming for carbon-aware cloud provisioning policy.

References

[1] Amazon elastic compute cloud. http://aws.amazon.com/ec2/pricing/.
[2] Apache web server. http://httpd.apache.org/.
[3] M. Arlitt and T. Jin. A workload characterization study of the 1998 world cup web site. *IEEE Network*, 14(3), 2000.
[4] J. Bialek. Tracing the flow of electricity. *IEE Proceedings on Generation, Transmission, and Distribution*, 143(4), 1996.
[5] P. Clayton. *Fundamentals of Electric Circuit Analysis*. John Wiley and Sons, 2001.
[6] G. Cook and J. Horn. How dirty is your data: A look at the energy choices that power cloud computing. In *GreenPeace Int'l*, Mar. 2011.
[7] N. Deng and C. Stewart. Concentrating renewable energy in grid-tied datacenters. In *International Symposium on Sustainable Systems and Technology (ISSST)*, May 2011.
[8] Environmental Leader. Data centers power up savings with renewable energy. http://www.environmentalleader.com/2009/07/29/data-centers-power-up-savings-with-renewable-energy/.
[9] D. Gmach, Y. Chen, A. Shah, J. Rolia, C. Bash, T. Christian, and R. Sharma. Profiling sustainability of data centers. In *ISSST*, 2010.
[10] 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 *International Conference on Network and Services Management*, Oct. 2010.
[11] Green House Data: Greening the data center. http://www.greenhousedata.com/.
[12] O. Ibarra and C. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *Journal of the ACM*, 22(4), 1975.
[13] K. Le, O. Bilgir, R. Bianchini, M. Martonosi, and T. D. Nguyen. Capping the brown energy consumption of internet services at low cost. In *International Green Computing Conference*, Aug. 2010.
[14] C. Li, W. Zhang, C. Cho, and T. Li. Solarcore: Solar energy driven multi-core architecture power management. In *HPCA*, Feb. 2011.
[15] Z. Liu, M. Lin, and L. Andrew. Greening geographical load balancing. In *ACM SIGMETRICS*, June 2011.
[16] S. Martello and P. Toth. *Knapsack problems: algorithms and computer implementations*. Wiley and Sons, 1990.
[17] N. Sharma, S. Barker, D. Irwin, and P. Shenoy. Blink: managing server clusters on intermittent power. In *ASPLOS*, Mar. 2011.
[18] N. Sharma, J. Gummeson, D. Irwin, and P. Shenoy. Cloudy computing: Leveraging weather forecasts in energy harvesting sensor systems. In *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, June 2010.
[19] C. Stewart and K. Shen. Some joules are more precious than others: Managing renewable energy in the datacenter. In *Workshop on Power Aware Computing and Systems(HotPower)*, Sept. 2009.
[20] G. USA. Facebook status update: Renewable energy now. http://www.greenpeace.org/usa/news/facebook-update-renewable-ene.