# Adaptive Green Hosting†

Nan Deng
The Ohio State University
dengn@cse.ohio-state.edu

Christopher Stewart
The Ohio State University
cstewart@cse.ohio-state.edu

Daniel Gmach
Hewlett Packard Labs
daniel.gmach@hp.com

Martin Arlitt
Hewlett Packard Labs
martin.arlitt@hp.com

Jaimie Kelley
The Ohio State University
kelley.530@osu.edu

## ABSTRACT

The growing carbon footprint of Web hosting centers contributes to climate change and could harm the public's perception of Web hosts and Internet services. A pioneering cadre of Web hosts, called *green hosts*, lower their footprints by cutting into their profit margins to buy carbon offsets. This paper argues that an adaptive approach to buying carbon offsets can increase a green host's total profit by exploiting daily, bursty patterns in Internet service workloads. We make the case in three steps. First, we present a realistic, geographically distributed service that meets strict SLAs while using green hosts to lower its carbon footprint. We show that the service routes requests between competing hosts differently depending on its request arrival rate and on how many carbon offsets each host provides. Second, we use empirical traces of request arrivals to compute how many carbon offsets a host should provide to maximize its profit. We find that diurnal fluctuations and bursty surges interrupted long contiguous periods where the best carbon offset policy held steady, leading us to propose a reactive approach. For certain hosts, our approach can triple the profit compared to a fixed approach used in practice. Third, we simulate 9 services with diverse carbon footprint goals that distribute their workloads across 11 Web hosts worldwide. We use real data on the location of Web hosts and their provided carbon offset policies to show that adaptive green hosting can increase profit by 152% for one of today's larger green hosts.

## Categories and Subject Descriptors

C.4 [**computer systems organization**]: performance of systems; H.1 [**information systems**]: models and principles

## Keywords

system management, green computing, web hosting, datacenter, renewable energy, performance and cost models, autonomic

†*Supported in part by NSF EAGER grant CNS-1230776.*



## 1. INTRODUCTION

Web hosting centers, or datacenters, provide fast servers at low cost for Internet services ranging from search engines to e-commerce sites. Their success has changed the world, but their growing carbon footprint is a concern. By 2020, the annual carbon footprint of hosting centers worldwide is expected to exceed the footprint of the entire Netherlands [27]. Such a large footprint would add to climate change and put Web hosts at risk of costly, punitive regulations. Web hosts can slow the growth of their carbon footprint by using less electricity from coal and other dirty energy sources. There are two ways to do this: 1) use less energy per hosted service and 2) use energy from clean sources in place of energy from dirty sources. The first approach, using less energy, not only reduces carbon footprint, it also reduces the cost of Web hosting and has been widely studied [9, 32, 34, 36]. The second approach, using cleaner energy, can reduce or even eliminate carbon footprints, but typically increases energy costs. Green hosts[1] —an emerging cadre of Web hosts that invest in costly, clean energy while maintaining low prices—must host more services than traditional hosts to profit from their investment. This paper helps green hosts invest efficiently, buying clean energy only if it will bring in new customers and increase profit.

We propose *adaptive green hosting*, the hosting center's response to geographically distributed Internet services [28, 31, 32, 46]. Such services meet their cost, SLA, and carbon footprint goals by routing their workload to hosts that offer either 1) low carbon footprints, 2) high performance, or 3) a little bit of both. Our key insight is that a green host can entice a service to route workload to it by providing more carbon offsets. In this paper, we use the term *carbon offset* to represent a unit of clean energy that can replace a unit of dirty energy, both measured in Joules. Carbon offsets can be produced by on-site solar panels, power received from local wind farms, or renewable energy credits (RECs) purchased via energy markets. In adaptive green hosting, Web hosts set the ratio of carbon offsets to dirty energy, henceforth the *offset ratio*, by observing their profit from each hosted service under various settings over time. Where geographically distributed Internet services look for hosts with good offsetting policies, adaptive green hosting sets policies to entice service providers to use them.

Adaptive green hosting contrasts with the approach most widely used in practice today, fixed offset ratios. In choosing a fixed offset ratio, today's green hosts try to meet the carbon footprint goals

[1]The term "green Web host", instead of green datacenter, is widely used in popular press and on business websites [4, 22, 35]. In this paper, we follow this precedent.

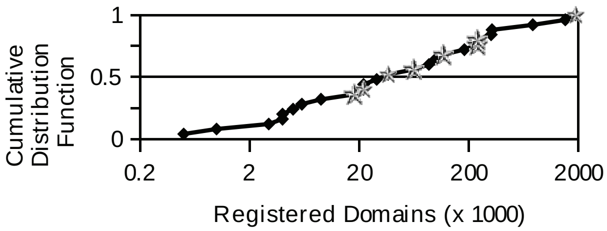



Figure 1: A niche market for green hosting. Each point represents 1 Web host. The X-axis is the number of A-type DNS records registered to the host. Stars indicate green hosts.

of their hosted services. However, meeting this threshold does not ensure that a green host will receive a service's workload. Instead, a service may route its workload across multiple hosts, mixing resources that differ in performance and offset ratios. This latter approach exploits fungible carbon offsets. The service needs only ensure that the weighted sum of their carbon footprint across all hosts meets their goals.

We compare fixed versus adaptive offset ratios using a geographically distributed service proposed in prior work [28]. The service chooses between 3 international Web hosts. We show that daily and bursty workload patterns alone can change the service's routing policies, suggesting that fixed offset ratios do not always maximize a green host's profit. At times, a fixed offset ratio causes green hosts to buy too many offsets (wasting money). At other times, it causes green hosts to buy too few offsets (losing customers). Our proposed adaptive approach can increase profit by 152% for one of today's larger green hosts. This paper makes the following contributions:

1. We present a new problem that intersects autonomic and green computing: Adaptive management of carbon offsets in shared web hosting centers.

2. We show that a green host's profit from an investment in clean energy is affected by the workload patterns of hosted services.

3. We propose a reactive approach to set the offset ratio that increases profit across diverse workloads and carbon footprint goals.

The remainder of this paper is as follows. Section 2 describes the state of the art in green hosting and discusses prior systems research on adapting to changing workloads. Section 3 overviews adaptive green hosting and defends its key premise: Time-varying workload changes can affect the yield of a given offset ratio. Section 4 computes a host's best offset ratio across real traces with diurnal and bursty patterns and proposes a reactive approach to set the offset ratio. Section 5 studies adaptive green hosting across multiple, geographically distributed services, showing that adaptive green hosting increases profits compared to an aggressive fixed offset ratio. Section 6 concludes by framing these results in the broader context of green and autonomic computing.

## 2. RELATED WORK

This paper models a green host's profit under adaptive and fixed carbon offset to dirty energy ratios, advancing the state of the art. Our models study how carbon-aware services react as green hosts increase their offset ratios, extending recent research on carbon awareness. Finally, we propose a reactive approach to set offset ratios, adding to a large body of work in workload adaptation. In this section, we outline related work in these areas.

**State of the Art in Green Hosting:** AISO [4], HostGator [22], Green Geeks, and GreenQloud [19] reflect a growing cadre of green hosts that hope to profit from their investments in clean energy. AISO, the eldest of these green hosts, was founded in 1997 but its customer base began to grow rapidly in 2002, increasing by 60% through 2008 [45]. AISO's growth marks the start of an ongoing boom in green hosting. HostGator [22], a green host based in windy Texas and founded in 2002, is now one of the largest low-cost Web hosts in the world, hosting over 1.8 million domain names. While AISO buys solar panels to invest in clean energy, HostGator buys renewable energy credits from local wind farms. The latter approach, using renewable energy credits, allows HostGator to support offset ratios greater than 100% by buying multiple credits for every joule used. HostGator in particular offsets 130% of the dirty energy used to power its servers. Green Geeks offset 300%.

Green hosting firms are targeting a small but growing market, Internet service providers that want show their commitment to the environment. Most people worldwide (83%) say that they prefer green products when they do not cost more than non-green alternatives [21], perhaps reflecting conspicuous altruism [20]. Similar results show that CIOs (61%) and system managers (71%) are willing to support green hosts if prices, response times, and throughput are the same [8, 23].

Figure 1 provides evidence of the growth. We plot registrations on the domain name services (DNS) of Web hosts, a rough but widely used metric to size Web hosts. Hosts with more authoratative (A-type) DNS records likely support more services. Using Domain Tools [12], we counted the DNS records of 200 Web hosts returned from online searches, plotting the 25 largest. In this group, there were 8 hosts that mentioned clean energy investments on their public Web pages (green hosts) and 17 traditional hosts that did not. We also controlled for price and hosting features. Each host offered hosting plans below \$5, a 99.9% uptime guarantee, and unlimited network data transfer. Most green hosts in our study were above the median in terms of registered domains with 2.9 times more A-type records than traditional hosts on average.

**Research on Carbon-Aware Services:** With the boom in green hosting, some services now consider their carbon footprint when choosing between Web hosts. These services, called geographically distributed services, can use many hosts worldwide. Le et al. [28] studied services that capped their carbon footprints either by cap-and-trade, cap-and-pay, or absolutely capped policies. Their key insight was that a central load balancer could route requests between green and dirty Web hosts to maintain a low carbon footprint while meeting SLAs. Liu et al. [31, 32] provided a model to assess a Web host's performance to carbon footprint efficiency. They use weighted linear models to find the best host, proposing a scalable algorithm to do so. Zhang et al. [46] studied services that tried to minimize the carbon footprint of certain requests within a fixed budget. This approach reflects a common practice where large companies outsource a small portion of their operation to a green host, often for conspicuous altruism [4, 20]. Ren et al. [38] also discuss the costs of going green.

Internet services can also consider their carbon footprint by deciding when, if ever, to process requests. A service can drop requests and turn off machines to use less dirty energy. Blink [39] proposed a key-value storage service that transferred popular keys away from nodes that were turned off during intermittent clean energy outages. The challenge was to serve as many read and write requests as possible using only resources powered by clean energy. Li et al. [29] turned off processor cores to increase the ratio of renewable energy to dirty energy on a system. Similarly, Gmach et

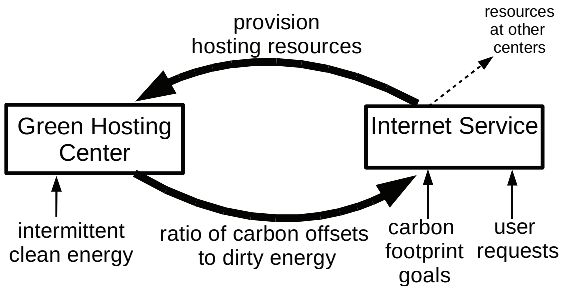



Figure 2: Adaptive green hosting for Internet services that lease resources on demand.

al. [16, 17] found that server-power capping and consolidation to power servers under low renewable-energy production can enable renewable powered services, albeit with a performance cost. Stewart et al. [42] was among the first to explore these problems, "showing that datacenters must use costly batteries or grid ties to make up for below-threshold renewable-energy production.

**Workload Adaptation:** It is well known that Web hosts support dynamic workloads that exhibit daily [5, 40], bursty [7], and non-stationary patterns [40]. Control theory solutions are now widely used in research and practice. Abdelzaher et al. [3] provides a good primer on such techniques, covering resource and admission controllers, sensors, and reactive and predictive techniques. There has been too much work in this area to list here. Our work adds to the field by considering a new metric, the offset ratio.

# 3. MAKING THE CASE FOR AN ADAPTIVE APPROACH

Given the boom in green hosting, we believe future Web hosts will compete by offering fast, cheap, and *green* resources to nimble cloud-based services. We propose adaptive green hosting, a new control loop based on carbon offset ratios (shown in Figure 2). Hosts adapt their offset ratios for each hosted service in response to changes in the availability of carbon offsets and request arrival patterns. We define a *carbon offset policy* as a vector where each element indicates the offset ratio assigned to each hosted service. This section makes the case for an adaptive approach by showing that fixed policies yield below optimal profits even when carbon offsets are always available at a fixed price and hosted services have fixed carbon-footprint goals.

## 3.1 Motivating Example

Consider Ecosia [13], a simple Internet service that provides a wrapper to Bing's search APIs and uses ad revenue to 1) offset Bing's estimated footprint and 2) invest in a rainforest protection program. Rather than spending its ad revenue on carbon offsets for the servers that host its homepage, CSS style sheets, and CGI scripts, Ecosia uses green hosts, bundling the costs of carbon offsets with hosting expenses. Ecosia commits to a carbon neutral footprint for its servers [13], i.e. 100% offset ratio. That is, Ecosia must be able to attribute 1 carbon offset for every joule of dirty energy used to power its servers. Every month these servers support more than 15 million unique searches that must complete quickly or else Ecosia will lose users [2].

For this example, we assume that Ecosia can send search requests that originate in the East Coast of the US to a Web host in either 1) the Eastern US, 2) the Western US, or 3) Europe. This setup mimics prior work [11, 28]. The hosts differ only in their network latency and carbon offsets per joule. The eastern host has the lowest network latency (41ms round trip on average), then the western host (80ms), and finally the European host (121ms). Each host leases cloud instances that can service a request in 1.6ms, supporting up to 600 requests per second (RPS). However, successful requests must complete within 150ms, including network latency, queuing delay, and service time. The expected successful requests from each datacenter is shown below, using an modified M/M/1 queuing model [25].

$$\text{Eastern US Host} \qquad v_0 = \frac{(0.150 - 0.041)}{\frac{1}{(600-\lambda_0)}} \tag{1}$$

$$\text{Western US Host} \qquad v_1 = \frac{(0.150 - 0.080)}{\frac{1}{(600-\lambda_0)}} \tag{2}$$

$$\text{European Host} \qquad v_2 = \frac{(0.150 - 0.121)}{\frac{1}{(600-\lambda_0)}} \tag{3}$$

Here, $\lambda_0$ reflects the request arrival rate at time 0. The eastern host offers no carbon offsets, the western host is carbon neutral, and finally the European host buys 2 offsets for every joule it uses. In other words, the hosts have offset ratios of 0%, 100%, and 200% respectively.

Ecosia wants to use as few cloud instances as possible while ensuring 1) all arriving requests complete successfully and 2) carbon footprint goals are met. Cloud instances are leased hourly. We assume that at every 1-hour interval $t$, Ecosia knows its request arrival rate for that interval, e.g., $\lambda_t = 120$ requests per second. With the request arrival rate, we can compute how many requests each instance can complete successfully (i.e., $v_0 = 51$,$v_1 = 38$,$v_2 = 16$ under $\lambda_t = 120$ RPS). Knowing the offset ratio for each instance (i.e., $c_0 = 0\%, c_1 = 100\%, c_2 = 200\%$) and Ecosia's goal of being carbon neutral ($C = 100\%$), we can compute the Ecosia's optimal workload distribution, i.e., the vector $X = \langle x_0, x_1, \cdots, x_i \rangle$ where each element reflects how many instances (an integer) Ecosia leases from each host $i$. The formal optimization model is:

$$\text{Minimize} \sum_{i=0}^{n} x_i^{(t)} \tag{4}$$

$$\text{Subject to} \frac{\sum_{i=0}^{n} E_i c_i^{(t)} x_i^{(t)}}{\sum E_i x_i^{(t)}} \geq C \tag{5}$$

$$\text{and} \sum_{i=0}^{n} v_i x_i^{(t)} \geq \lambda \tag{6}$$

$$\text{and } \forall_i (x_i \in \mathbb{Z}) \tag{7}$$

The goal is to minimize the total number of instances used. The first constraint keeps Ecosia's servers within a target carbon footprint (C). To be carbon neutral, Ecosia would set $C = 0$. Assuming green hosts and traditional hosts differ only in their offset ratio, we uniformly set the energy per instance coefficient ($E_i$) to 100wH. The second constraint requires enough instances to process incoming requests ($\lambda_t$) within SLA.

Integer programming solvers can find near optimal workload distributions for Ecosia [28,31,46]. We used LP solve, an open source solver commonly bundled with Linux platforms [33]. Under 120 RPS, Ecosia would use 4 instances from the host in western US only. Even though the host in eastern US can successfully complete 1.3X more requests per instance, the lack of carbon offsets forces Ecosia to use other hosts.

Under adaptive green hosting, the eastern host could buy carbon offsets specifically to attract Ecosia's workload. The carbon-offset

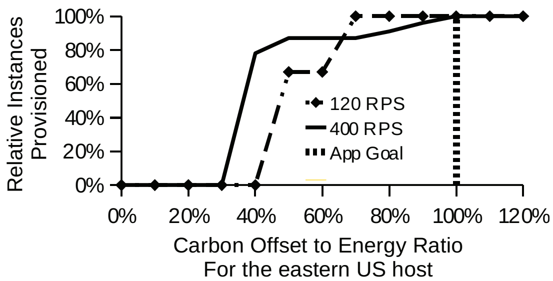



Figure 3: Carbon-offset elasticity for the eastern US host. The y-axis shows instances provisioned on the host relative to the maximum setting, i.e., $\frac{\eta_{east}(X)-\eta_{east}(K)}{\eta_{east}(K)}$ where K maximizes $\eta_{east}$. Ecosia routes requests differently across offset ratios (x-axis). Under 120 requests per second (RPS), $\eta_{east}(K)$ equals 3 instances. Under 400 RPS, it equals 23 instances.

elasticity ($\eta$) captures a host's workload as a function of carbon offsets assigned ($\zeta$) to a target service. The carbon-offset elasticity tells us if a host can increase its workload by giving a target service more offsets per joule of dirty energy. These offsets can be bought as renewable energy credits, transferred from another service, or pulled from on-site sources. Because energy is fungible, this is an accounting problem. Below, we show the optimization formula for carbon offset elasticity for a single service. Equation 9 projects Equation 5 to a single host that considers the marginal gain by changing its offset ratio (Equation 10).

$$\eta_j(\zeta) = x_j : \text{ Minimize } \sum_i^n x_i^{(t)} \tag{8}$$

$$\text{Subject to } \frac{\sum_{i=1}^n c^{(t)} x_i^{(t)} + \zeta x_0^{(t)}}{\sum x_i^{(t)}} \geq C \tag{9}$$

$$\text{and } \sum_{i=0}^n v_i x_i^{(t)} \geq \lambda \tag{10}$$

For $N$ discrete settings of $\zeta$, we can compute a host's carbon offset elasticity for a model-driven service by solving $N$ integer programming problems. *Thoughout this paper, we use this key insight to assess the yield of clean energy investments for a host.*

Figure 3 shows the carbon-offset elasticity for the eastern host. The result highlights a unique aspect of clean energy: it is fungible. Even though Ecosia managers want their service to be carbon neutral, they will lease instances from a host that offsets less than 100% of its carbon footprint if other hosts offset more than 100%. In this example, the eastern host benefited. Under 120 RPS, if the eastern US host were to offset just 50% of its carbon footprint, the best workload distribution used only eastern US and European instances. If the eastern US host were to offset 70% of its carbon footprint, the best workload distribution used 1 European and 3 eastern US instances, matching the the number of instances used if the host were to offset 100% (carbon neutral).

The carbon elasticity changes when Ecosia's request arrival rate rises to 400 RPS. Under 70% carbon offset ratio, European instances detracted 13% of the workload that would be sent to the eastern US host if it were carbon neutral. In fact, under 400 RPS, the eastern host leases the same number of instances under a 50% offset ratio as it does at the 70% offset ratio. This shows that a static carbon offset policy chosen under 1 request arrival rate can be below optimal when the request rate changes. Note, this finding does not require that Ecosia managers change their carbon footprint goals or relax their SLA, nor does it require that carbon offsets become more or less available. Also, we observe that the elasticity function grew slowly after 40% offset, raising the question, "does a 15% increase in leased instances justify a 60% increase in the offset ratio?" We address this question Section 4.

Table 1: A summary of all outcomes for the workload distribution found via integer programming solution for carbon-capped and performance-oriented services.

| *Conditions* | *Hosts chosen* |
|---|---|
| $c_{BP} \geq C$ | East |
| $\forall_i lp_i(\lambda) \in Z$ and | East, West, or Euro |
| $\sum v_i \cdot [lp_i(\lambda) - \lfloor(lp_i(\lambda))\rfloor] \leq 2v_{sp}$ | East, West, or Euro |
| $\sum v_i \cdot [lp_i(\lambda) - \lfloor(lp_i(\lambda))\rfloor] > 2v_{west}$ **and** $\sum v_i \cdot [lp_i(\lambda) - \lfloor(lp_i(\lambda))\rfloor] \leq v_{east}$ | East, West Euro, $Euro_\infty$ |
| $\sum v_i \cdot [lp_i(\lambda) - \lfloor(lp_i(\lambda))\rfloor] > 2 * v_{east}$ **and** $\sum v_i \cdot [lp_i(\lambda) - \lfloor(lp_i(\lambda))\rfloor] \leq v_{east} + v_{euro}$ | East, West Euro, $Euro_\infty$ |

## 3.2 Generalizing the Example

The relative throughput and offset ratios of the hosts in our example capture a practical region of the workload distribution problem for carbon aware services. The general problem is an integer programming problem; each service assigns an integer ($ip_i \in \mathbb{Z}$) to n-tuples $(v_i, c_i)$ reflecting the instances leased from each host. The ideal solution is not limited by the integer requirement and finds a solution equal to the linear programming solution ($lp_i \in \mathbb{R}$). We constrain this space of problems with the following assumption: a service will consider only 1 host that doesn't meet its carbon footprint goals, the best performing host. Our assumption builds from the intuition that workload distribution involves some management costs that will deter managers from choosing poor performing hosts that offer too few offsets to meet a service's goals. We call services that follow this assumption performance oriented.

We claim that any carbon-capped and performance-oriented service will lease instances from only 1) its best performing host, 2) its best performing host that meets carbon footprint goals (i.e., second best performing host), or 3) the host offers an offset ratio that exceeds the service's footprint goals and combines with the best performing host to yield highest performance per instance achievable while meeting footprint goals (exploiting fungible offsets).These properties correspond to the eastern US, western US, and European hosts in the Ecosia example. Changing the absolute throughput and offset ratios of a service's hosts will change the proportions with which each host is selected. However, our claim is that any host used by a carbon capped service will have (in the limit) at least one of the 3 properties above.

We prove this claim by considering all possible outcomes of the optimization model. Table 1 provides a summary of our proof.

**Outcome #1: The best performing host offers an offset ratio that exceeds the service's carbon footprint goals.** The service uses instances from only the eastern US host.

**Outcome #2: The linear programming solution returns only integer values.** With only 2 constraints, an n-host linear programming solution chooses between only 3 hosts [43]. The service picks instances from either only the western US host or some linear combination of the eastern US and European hosts, whichever provides the best performance per instance. The outcomes here are restricted by our prior assumption that each service considers at most 1 dirty host. We leave to future work an extension of this analysis for services that can use more than 1 dirty host. For such services, any mix of dirty and green hosts could be the most efficient, which would make computing the $\eta$ function more complex.

**Outcome #3: The linear programming solution uses fractional instances to process fewer than $2v_{west}$ requests.** Here, the integer programming solution replaces the fractional instances with whole instances. The western US host represents the most efficient way to do this, since, by definition, it offers the greatest performance among hosts that meet carbon footprint goals.

**Outcome #4: The linear programming solution uses fractional instances to process fewer than $v_{east}$ requests.** Here, we rely on the performance-oriented assumption. The service either provisions (more than 2) instances from only the western US host, or it mixes instances with the eastern US host and some other host that exceeds its footprint goals. The service must use either the eastern US host or the western US host because no other host offers fewer offsets than the western US host and exceeds its throughput. As the Europoean host's offset ratio goes to infinity, we can show that it becomes the host that the eastern US host is combined with. Thus, we denote it as $euro_{\infty}$ in Table 1.

**Outcome #5: The linear programming solution uses fractional instances to process fewer than $v_{east} + v_{euro}$ requests.** This outcome combines instances from Outcome #3 and #4. Finally, we note that the linear programming solution would not process more fractional requests than $v_{east}$ + $v_{euro}$.

# 4. ADAPTING TO REAL WORKLOADS

Section 3 described an Internet service that divided user requests among competing hosts to 1) be carbon neutral and 2) keep its costs low. Hosts received a portion of the service's requests, depending on their cost to throughput ratio, carbon footprint, and the rate at which user requests arrived. This example showed that, as request rates change over time, green hosts that use fixed offset ratios will sometimes lower their profit by buying too many (spending more than needed) or too few offsets (losing customers).

This section shows that green hosts can increase profit derived from a service by eschewing fixed offset ratios in favor of an adaptive approach. Prior research on adapting to workload changes has focused on how services should provision instances to maximize throughput [3], minimize costs [15, 32], and meet carbon goals [28, 31, 46]. In this section, we focus on how *hosts* should set their offset ratio (e.g., by buying RECs) to maximize their profit for a service. Like prior work, this function depends on the service's request rate, cost models, and carbon footprint goals. However, unlike prior work, this function also depends on the performance and offset ratios of other hosts.

We revisit our example service from Section 3. This time, we use a trace from a real enterprise service to capture changing request rates. For each 1-hour window in the traces, we compute carbon offset ratios that maximize profit for the eastern, western, and European hosts. We study 1) how many times the best carbon offset ratio changes, 2) how quickly it changes, and 3) how much it changes. Our results prompted us to create a reactive approach that adapts the offset ratio based on recent history. We begin by presenting a formal profit model for green hosting.

## 4.1 Profit Model

Web hosting centers that adopt a cloud computing model earn money by leasing virtual resources over a fixed period of time [6]. A leasable resource is called an instance. Hosts profit when they earn more money per leased resource than they spend buying, maintaining, and powering them (captured in Equation 11).

$$P = I \cdot p \cdot R - \frac{StartupCosts}{T} \tag{11}$$

In the above equation, profit $P$ is a function of instances leased ($I$), revenue per instance ($R$), the percentage of revenue turned into profit considering only operational costs ($p$), and amortized startup costs (where $T$ captures the host's expected lifetime). We assume $I \geq 1$. In most places, clean energy costs more than dirty energy, so green hosts will have higher operational costs. They must lease more instances to profit from this investment.

$$P(c) = \eta(c) \cdot p \cdot R - c \cdot E \cdot cost_{co2e} \cdot S - \frac{StartupCosts}{T} \tag{12}$$

$$P(c) = \eta(c) \cdot p \cdot R - c \cdot E \cdot cost_{co2e} \cdot \lceil \frac{\eta(c)}{S} \rceil S - \frac{StartupCosts}{T} \tag{13}$$

Equation 12 adds the cost of carbon offsets ($cost_{co2e}$), energy per instance ($E$), the granularity of energy data (measured in instances) ($S$), and the ratio of carbon offsets to joules ($c$). These factors make green hosting less profitable than traditional Web hosting. The equation also shows the effect of carbon offset elasticity ($\eta(c)$) in increasing the amount of instances leased. Green hosts can profit by investing in clean energy only when the carbon offset elasticity leads to increased revenue. Equation 13 shows the full profit model when $\eta(c)$ can exceed S.

In practice, green hosts invest in clean energy with caution, trying to keep the risk of losing money low. Here, we formalize a risk aware approach commonly used in practice [14, 26]. The idea is to cap how much money is invested in clean energy so that a small increase in leased instances yields profit.

**Low Risk Green Hosting:** *The maximum ratio of carbon offsets to dirty energy ($c_{max}$) is capped, such that $c_{max} \leq \frac{pR}{E \cdot cost_{co2e} \cdot |S|}$. Where $S$ is the set of leasable instances receiving the offsets. Plugging $c_{max}$ into Equation 12, we see that it allows a host to recoup costs when increasing the offset ratio from 0 to $c_{max}$ yields only 1 leased instance (the worst case).*

Theorem: A green host that invests with the above low-risk approach should choose the smallest $c$ that maximizes $\eta(c)$ in order to maximize profit. Here, we provide a short proof. First, we observe that a host's costs are linear in $c$, provided $E > 0$ and $cost_{co2e} > 0$. If $\eta(c + \epsilon) = \eta(c)$, then costs under $c + \epsilon$ would exceed costs under $c$, meaning lower total profit. Thus, the smallest $c$ is a necessary condition. Second, we prove by contradiction that $\eta$ must be maximized.

$$Hypothesis : Assume P(c_1) > P(c_2) \text{ where } \eta(c_1) < \eta(c_2) \tag{14}$$

$$WLOG : StartupCosts = 0 \tag{15}$$

$$Substitution : P(c_1) = \eta(c_1) pR - c_1 E cost_{co2e} |S| \tag{16}$$

$$Substitution : P(c_2) = \eta(c_2) pR - c_2 E cost_{co2e} |S| \tag{17}$$

$$WLOG : Assume c_1 = 0 \tag{18}$$

$$Substitution : \eta(0) pR > \eta(c_2) pR - c_2 E cost_{co2e} |S| \tag{19}$$

$$Algebra : \frac{c_2 E cost_{co2e} |S|}{pR} > \eta(c_2) - \eta(0) \tag{20}$$

$$WLOG : Assume \eta(c_2) - \eta(0) = 1 \tag{21}$$

$$WLOG : Assume c_2 = c_{max} \text{ i.e., as large as possible} \tag{22}$$

$$Contradiction : \frac{|S|}{|S|} > 1 \tag{23}$$

Finally, we used both public data and local tests to calibrate a realistic $c_{max}$. Table 2 shows inputs to our profit model and their

Table 2: Values used to estimate $c_{max}$ for this study.

| Public Data | | |
|---|---|---|
| variable | value | source |
| $R$ | \$0.085 | Amazon EC2 [1] |
| $p$ | 4% | Amazon's EBITDA [44] |
| $cost_{co2e}$ | \$0.0045 | Renewable energy credits online [18] |
| Local Tests | | |
| variable | value | source |
| $E$ | 23Kj | ARM Marvel processor + SSD |
| $S$ | 32 | Tripp Lite PDU with power display |

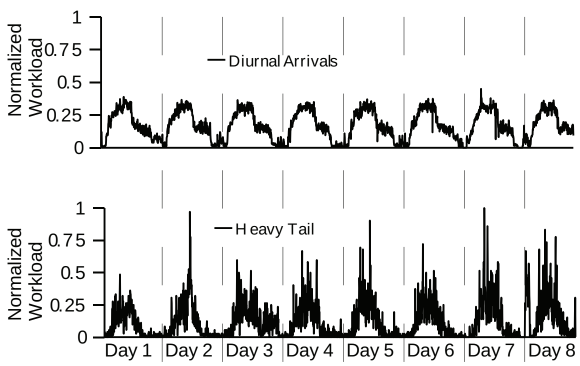



Figure 4: Request rates for a modern enterprise application, codenamed VDR [40]. VDR is used in six continents. The plots show requests rates at 2 servers hosted in the Americas. The first plot compiles arriving requests for both servers, capturing diurnal patterns. The second plot shows request rates for a request type with fast response times, likely static content. In the second plot, requests arrive according to a heavy tail.

source. Our local setup uses a small cluster of ARM processor devices with attached SSD storage. These devices host Apache on Linux, supporting up to 600 requests per second throughput. These results match findings from prior work [30, 37]. The peak power from our ARM nodes is 5.5W; multiplying by 3,600 seconds provides our value for the hourly energy usage of an instance. We also consider a PUE of 1.2. Most (81%) power distribution units (PDU) used at the rack level in today's datacenter include LCD displays and network access for energy data [41]. Our PDU can support 32 instances. Note, the PDU is a good level to assign carbon offsets since energy data is easy to acquire. Assigning carbon offsets at higher levels in the power delivery system increase the size of S, diluting the amount of carbon offsets that can be purchased with low risk [10]. Filling these values into our model, we set $c_{max} = 300\%$ for all studies in the remainder of this paper.

## 4.2 Trace-driven Study

We used empirical traces of request rates and carbon prices to study the most profitable carbon offset ratios for green hosts over time. Recall, in Section 3, we computed $\eta$ for the eastern host using a constant request rate and the default offset ratios of the western US and European hosts. In this section, we compute $\eta$ for $T$ timestamped request rates and offset ratios. Assuming low risk investing, the output reduces to a vector of $T$ carbon offset ratios for each host, where the $t^{th}$ setting reflects the smallest ratio $c_t$ that maximizes $\eta_t(c_t)$ given the request rate $\lambda_t$. Our final assessment of profit uses our model to combine results from all $T$ time steps.

Figure 4 shows two normalized request rate traces taken from an HP service used across the world [40]. These traces cover approx. 8 days and capture diurnal patterns in the request rate. Both traces were normalized to produce about 1.5 million requests per day (about 175 RPS). They differ in the distribution of request rates within a day. The top trace matches the distribution of all arriving requests. Its $99^{th}$ percentile of request rates is 1.5X larger than the $99^{th}$ percentile of an exponential distribution with the same mean. In other words, the top trace has a tail that is only slightly heavier than an exponential distribution. The bottom trace captures the arriving requests for 1 request type. The $99^{th}$ percentile of request rates in this trace is 5X larger than the $99^{th}$ percentile of a normal distribution with the same mean. In other words, the bottom trace has a tail that is much "heavier" than an exponential distribution. Such heavy tails are a well studied in Internet services [5, 7].

We also studied the effect of changing carbon prices by discounting $c_{max}$ and default offset ratios. We used a trace of the daily market price for carbon offsets from iPath Global Carbon [24]. Our trace ranged from Feb. 8, 2012 through Feb. 14, 2012. The resulting daily, relative prices were 1.08, 1.08, 1.03, 0.98, 0.97, 0.94, and 1. Market prices often track wholesale prices well.

**Study Results:** We used the iterative method described in Section 3 to compute carbon offset elasticities for each host, workload, and time step. We chose 31 discrete values for the offset ratio, using multiples of 10% from 0 to 300%. For every 1-hour time step, we used the request rate ($\lambda_t$) from either the diurnal or heavy tail traces above to compute how many instances a host would provision if it set its offset ratio to one of the above discrete values. We assume that the other hosts keep their default offset ratio.

Table 3 shows how the best carbon offset ratio changed over time under 1) the diurnal workload with fixed carbon prices, 2) the heavy tail workload with fixed carbon prices, and 3) the heavy tail workload with changing carbon prices. In total we computed 1,674 offset elasticities (3 hosts x 3 workloads x 186 hours).

To maximize profit across each studied workload, every host needed to use at least 3 different offset-ratio settings. The hosts used fewer settings (below 4) under the diurnal workload than under the heavy tail traces. We explain these results by highlighting a key aspect of the integer programming (IP) outcomes outlined in Section 3: Linear programming (outcomes #1 and 2) provide the best solution modulo the request rate. When the request rate is larger than $v_{east} + v_{euro}$, a host should set its offset ratio to maximize its usage under the linear programming solution. For the eastern and western US hosts, this setting is very close to the service's carbon footprint goal. For the European host, this setting is the smallest setting that ensures the following $k{*}v_{east}{+}v_{euro} > v_{west}$ where $k$ in $\mathbb{R}$ is the number of eastern US instances sponsored by the European host's fungible offsets. However, when the request rate falls below $v_{east} + v_{euro}$ (outcomes #3–5), the best settings change depending on the IP solution.

In our example, $v_{east} + v_{euro}$ equals 67 RPS and the average arrival rate is 175 RPS. The distribution of request rates in the diurnal workload is close to exponential, meaning the median request rate is close to the average rate. Indeed only 18% of the 1-hour intervals under the diurnal had request rate below 67 RPS. The offset ratio found under the linear programming solution was chosen of 90% of the time for all hosts. Heavy tail distributions do not share this property. Instead, short workload bursts make the average rate larger than the median. Despite having the same arrival rate, the heavy tail distribution shows request rates below 67 RPS 45% of the time. More generally, we can not claim that all heavy tail workloads on all cloud platforms will include some intervals where $\lambda_t < v_{east} + v_{euro}$. However, for a given average arrival rate, a high variance, heavy tail distribution is more likely than an exponential distribution to include such intervals.

Table 3: Data on the best carbon offset ratios in our study.

Metric: Number of Ratios Chosen ($|\bar{c}|$ : $c \in \bar{c}$ iff $\exists t\ \forall k\ P_t(c) \geq P_t(k)$)
What to look for: Numbers greater than 1 suggest that fixed policies yield below optimal profit.

| *Workload* | *Eastern* | *Western* | *European* |
|---|---|---|---|
| Diurnal | 4 settings | 3 | 3 |
| Heavy tail | 5 | 4 | 5 |
| Heavy tail and market carbon prices | 7 | 8 | 5 |

Metric: Expected Contiguity of $2^{nd}$ Most Frequent Setting (Ex(|L|: $\forall$ l $\in$ L $C_l = k \wedge$ (l+1) $\in$ L)
What to look for: Large numbers suggest carbon offset ratios are stable.

| *Workload* | *Eastern* | *Western* | *European* |
|---|---|---|---|
| Diurnal | 1.60 hours | 1.75 | 1.23 |
| Heavy tail | 1.65 | 3.00 | 4.05 |
| Heavy tail and market carbon prices | 2.05 | 2.75 | 8.86 |

Metric: Absolute distance from $1^{st}$
What to look for: Large numbers reflect the magnitude of profits lost by static hosts. Minimum value is 10% and maximum is 300%.

| *Workload* | *Eastern* | *Western* | *European* |
|---|---|---|---|
| Diurnal | 100% | 100% | 200% |
| Heavy tail | 100% | 50% | 190% |
| Heavy tail and market carbon prices | 100% | 100% | 100% |

Second, we observe that offset ratios change slowly. In particular, we observe several long contiguous periods under the second most frequent policy, even under the diurnal workload. Several last longer than 4 hours. We note that this correlated behavior is well explained by low request rates 1) at night and 2) between bursty periods. The average period of contiguity rounds up to 2 hours in all but 1 of the study traces. Finally, we also observe that the absolute distance between the second most frequent and most frequent ratio are far apart, simply setting the offset ratio to the larger of the two can waste a lot of money.

Table 4: Accuracy of reactive and tail-aware reactive approaches. Shown for the western US host.

Metric: Accuracy ($\sum_n \frac{I(C_n = Pred(n))}{n}$))

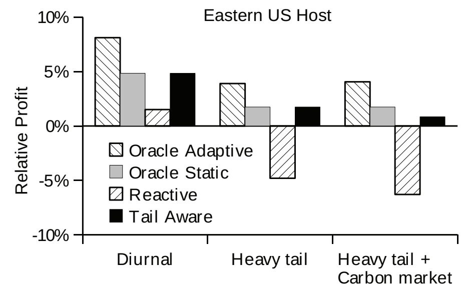
| *Web host* | *Mode* | *Reactive* | *Tail Aware* |
|---|---|---|---|
| Diurnal | 97% | 95% | 97% |
| Heavy tail | 66% | 65% | 70% |
| Heavy tail w/ Carbon market | 79% | 73% | 80% |

## 4.3 A Reactive Approach

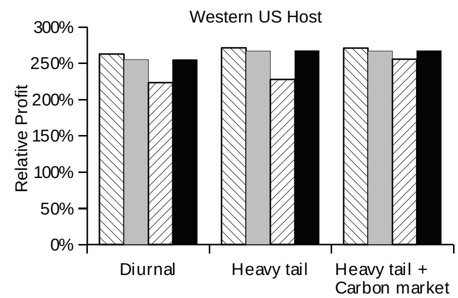
Since our trace-driven approach revealed that the best offset ratio held for long contiguous periods, we implemented a reactive approach to set the carbon offset ratio. We assume that Internet services tell each host what their ideal offset ratio was for the previous hour. Given that the service can monitor its request arrival rate, it can compute this offset directly using the approach described in Section 3.

Our reactive approach considers the history of a service's ideal offset ratio. When the ideal ratios over the last 2 hours match, we change the offset ratio to the matching value. Otherwise, we assign the ratio to the statistical mode. The latter works well under diurnal workloads where the most frequent ratio occurs 97% of the time. The former helps with heavy tailed workloads where the ratio changes for several hours at time.

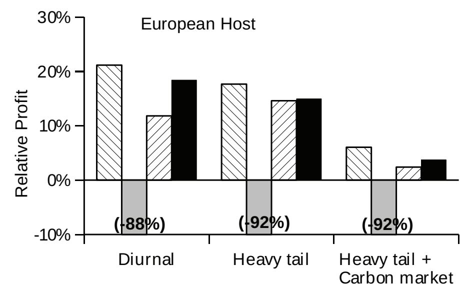
Our full approach also exploits heavy tailed contiguous periods in the offset ratio. We scan the history of results for patterns indicating that a contiguous period of length $l$ has a large probability of leading to a period of length $l + k$. If such patterns are found, our reactive policy returns to the mode after the $l + k$ interval. Table 4 shows the accuracy of our full, *tail-aware* approach in predicting the ideal offset ratio compared to using the mode or our base approach for the western US host. Our full approach makes the most accurate predictions in all cases, a property that holds across all studied hosts.




(a) Eastern US Host




(b) Western US Host




(c) European Host

Figure 5: Profit of east, west, and European hosts from the Ecosia example using real workload traces. All results are reported relative to the profit under the over-offsetting approach.

**Results:** We compare our reactive approach to an oracle-driven adaptive approach that sets the offset ratio to the value that maximizes profit for the upcoming interval (called *oracle adaptive)*). We also compare against an oracle-driven fixed-setting approach that sets the offset ratio to the value that most frequently maximized profit throughout the trace (i.e., the statistical mode for the whole trace). These approaches use advanced knowledge that would be unavailable in a deployed system, but they are useful in demonstrating how well our reactive approach works. We also compare against the *over offsetting* approach which sets offset ratio to $c_{max}$. The idea behind this approach is that increasing the offset ratio will only increase $\eta$ (which is not true). We make this over-offsetting approach our baseline.

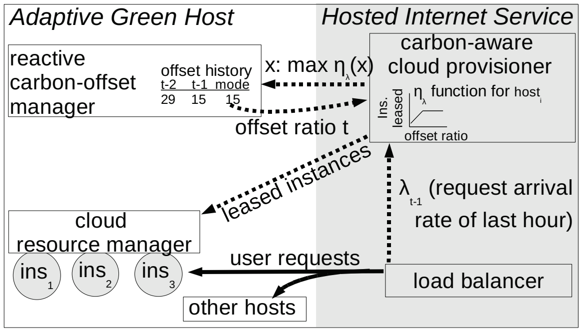
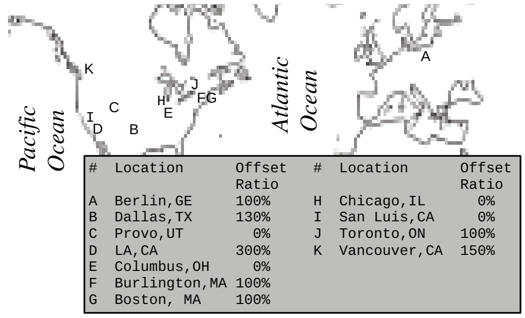



Figure 6: Our setup for adaptive green hosting. Dotted lines reflect data that is transmitted at every cloud provisioning interval (e.g., hourly). Solid lines reflect real time actions.




| # | Location | Offset Ratio | # | Location | Offset Ratio |
|---|---|---|---|---|---|
| A | Berlin,GE | 100% | H | Chicago,IL | 0% |
| B | Dallas,TX | 130% | I | San Luis,CA | 0% |
| C | Provo,UT | 0% | J | Toronto,ON | 100% |
| D | LA,CA | 300% | K | Vancouver,CA | 150% |
| E | Columbus,OH | 0% | | | |
| F | Burlington,MA | 100% | | | |
| G | Boston, MA | 100% | | | |

Figure 7: Where the shared Web hosts in our case studies live. We chose 11 of the largest Web hosts (green and traditional) using domain tools and online searches. Unintentionally, our results include hosts in North America and Europe only.

Figure 5 shows the profit achieved under the different carbon offsetting strategies. We highlight results from each host individually.

*1. The eastern US host* provides the best performance. If it raises its offset ratio about 100%, Ecosia would provision all instances on it. However, this may result in buying too many offsets. Generally, the profit loss from buying too many offsets, under low-risk investing, is lower than the loss from buying too few offsets. Thus, our base reactive approach which can under provision resources after noisy fluctuations in the ideal offset ratio is less profitable than the over-offsetting approach. Our tail-aware approach avoids losing money due noisy fluctuations by preemptively moving back to the mode when heavy tailed periods expire. We observe that small, short term changes in the ideal offset ratio can affect achieved profit.

*2. The western US host* sits in the middle. If the eastern host offers no carbon offsets, Ecosia often chooses the western host as a second option. However, since carbon offsets are fungible, if the western host increases its offsets to $c_{max}$, Ecosia will use the western host only so it can offset instances on the eastern host. Over-offsetting performs poorly. Comparing only the other approaches, the results are similar to the eastern US host.

*3. The European host* provides qualitatively different results. This host must offer high offset ratios (near $c_{max}$) to entice Ecosia to use it in combination with the eastern US host. It is most often best for this host to set an offset ratio of 0% to avoid wasting money on instances that won't be provisioned. However, for long periods in the middle of the day, the European host is best served by an offset ratio above 200%. Recall from Figure 3 that under high requests rates the European datacenter can reduce the offsets needed to provision eastern US instances. The mode is the wrong metric for this datacenter because the lost profit in the middle of the day far outweighs the cost of over offsetting. Our full approach is comparable to the optimal adaptive approach for this host.

# 5. CASE STUDIES ON SHARED HOSTS

Figure 6 details adaptive green hosting. At every provisioning interval, the hosted service uses recently observed data on its request arrival rate to compute the offset ratios that would maximize instances leased from each host (Section 3). The adaptive green hosts keeps a history of such data, and uses it to set its offset ratio for the next interval (Section 4). The hosted service then tries to maximize throughput within a carbon budget based on each host's performance and offset ratio by balancing its workload across hosts. In Section 4, we studied the effect of adaptive offset ratio on one service, finding that green hosts can increase profit by adapting their offset ratio to the service's daily and bursty workload patterns. This section studies hosts that support many services.

## 5.1 Setup

We used our VDR traces to simulate 9 Ecosia services. Each service used a load balancer to route requests to either: 1) its best performing host, 2) the best performing host that met its carbon footprint goals, or 3) a host that offered a high offset ratio. We defined these services such that the best performing host mapped to one of the large Web hosts described in Figure 1.

Each service placed its load balancer at the best performing host and set its carbon footprint goal to the offset ratio of the fastest green host. When the load balancer sent requests to a remote host, the penalty was 1 round trip network delay (as in the queuing models in Section 3). We modeled delay between hosts using: 1) distance in miles between the other hosts and the nearest host, 2) speed of light, 3) a slowdown coefficient, and 4) TCP processing overhead. We calibrated the slowdown coefficient with regression tests on ping results between a laptop in Columbus, OH and servers deployed in London, UK, Frankfurt, GE, Berkeley, CA, St. Louis, MO, and Rochester, NY. We set the coefficient to $2.4$.

Figure 7 plots the cities where each Web host's servers resided. The legend in the figure shows the carbon offset to dirty energy ratio offered by each host. We collected this data from public websites. There are 11 hosts listed, each is labeled with a letter to hide its identity. The two hosts offering the most carbon offsets (D and K) do not provide the highest throughput for any service.

Table 5 shows the set up for each service's load balancer and its carbon footprint goal. Two hosts (B and K) that offered offset ratios greater than 100% were used by services with diverse carbon footprint goals. Also, one well located carbon-neutral host (J) supported diverse footprint goals. Specifically, host B supported 7 services with the following goals: 100%, 100%, 130%, 130%, 130%, 130%, and 150%. Host J supported 3 services with the following goals: 100%, 100%, and 130%. Finally, host K supports 6 servies with the following goals: 150%, 130%, 130%, 130%, 130%, and 130%. We used the heavy tailed VDR trace for each service (Figure 4). The price of carbon offsets was fixed. The maximum throughput of each node was 600 requests per second.

At the top of every hour, our tail-aware reactive approach collected the ideal offset ratio for each service during the previous hour. We set the offset ratio for each service individually. Total profit for a host was the sum of profit from each hosted service. We compared this approach to the fixed offset policies commonly used in practice: 100%, 150%, 200%, and 300%. Here again, we call 300% the over offsetting approach and used it as our baseline.

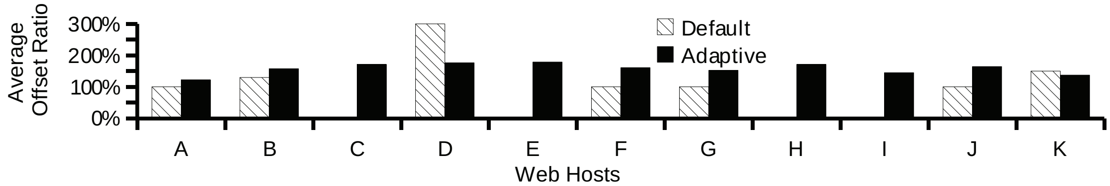



Figure 9: Average offset ratio recommended by adaptive green offsetting for each host in our setup.

Table 5: The configuration of each service's load balancer in our setup. The leftmost columns show the service number and its footprint goal. The rightmost columns label which hosts the service routes requests to.

| # | footprint goal | Best performing | Best performing + meets goals | Many Offsets |
|---|---|---|---|---|
| 1 | 100% | E | J | B |
| 2 | 150% | C | K | D |
| 3 | 100% | H | J | B |
| 4 | 130% | A | B | K |
| 5 | 130% | J | B | K |
| 6 | 130% | F | B | K |
| 7 | 150% | I | K | D |
| 8 | 130% | G | B | K |
| 9 | 130% | B | K | D |

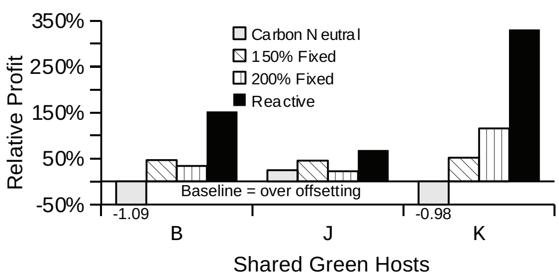



Figure 8: Relative profit of the shared green hosts (B, J, and K). Each host's profit per service under the over offsetting policy was \$2.17, \$7.66, and \$1.5 respectively. We used the VDR request trace with heavy tail arrival patterns (7.8 days). The over offsetting policy sets a fixed offset ratio of 300%. Recall, only hosts B, J, and K were shared by services with diverse footprint goals.

## 5.2 Shared Hosting Results

Figure 8 shows the relative profit increase from our *adaptive green hosting approach*. Our approach consistently outperformed the over offsetting approach, increasing profit by at least 68% in each case. Our gains were lowest (68%) for host J because its hosted services saw a wide difference in the offset ratio between their best performing hosts (0%) and host J (100%). Any investment in carbon offsets offered high yield. Indeed, the profit per service under the over offsetting policy (\$7.66) was 2–4 times larger than the other hosts. Here, our approach increases profit by adapting to workload changes in services. We also run the same experiment on diurnal traces mentioned in Section 4. The relative profit increase for host B, J and K are 105%, 69% and 236% respectively.

We also compared two approaches commonly used in practice: over offsetting and carbon-neutral green hosting. Host B and K gain the most from over offsetting because they were in competition against other green hosts. Host J preferred a carbon neutral approach. First, our approach adapted to each host's environment, consistently outperfoming both approaches.

**Is Adaptive Green Hosting Really Green?** Adaptive green hosting increases profit in two ways. First, it helps green hosts buy

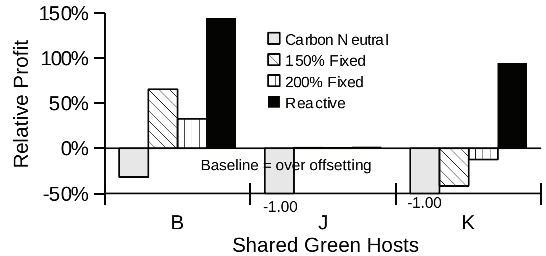



Figure 10: Relative profit of the shared green hosts when services provision according to a different optimization model [46]. Each host's profit per service under the over offsetting policy was \$6.33, \$5.50, and \$4.83 respectively.

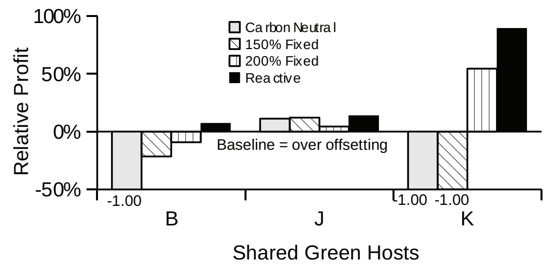



Figure 11: Relative profit of the shared green hosts when services can choose to buy carbon offset directly. Each host's profit per service under the over offsetting policy was \$1.71, \$2, and \$0.66.

carbon offsets with low risk, allowing them to make bold investments (up to $c_{max}$) to bring in customers. Second, it helps green hosts avoid wasting money on too many offsets. This latter benefit could actually make hosts less green than they are today. Figure 9 shows the suggested average offset ratio of adaptive green hosting in our setup. The average offset ratio increased for 10 of the 11 hosts. Only host D, which offered a ratio of 300%, had a lower average offset ratio than its default. Because green hosts reflect a minority of web hosts in general, adaptive green hosting is likely to suggest increased investment in clean energy.

**Can Adaptive Green Hosting be Applied to Different Service Models?** The services that we have studied so far have been based on minimizing instances (cost) within carbon and throughput constraints [28]. However, recent work has explored alternative models. Zhang et al. [46] proposed a model that maximizes renewable energy usage within cost and throughput constraints. Our approach to create carbon offset elasticity models can be applied to this service model also. We modified our setup to allow services #1, 2 and 3 to use this service model. Figure 10 shows the results. Services in this model tend to route a few requests to the greenest datacenter. Host J (which offers on 100% offset ratio by default) suffers the most. Over offsetting helps this host the most. Hosts B and K can adapt not only to supporting diverse carbon footprint goals but even to diverse service models. Our adaptive approach increases relative profit by more than 100 percentage points for both hosts.

**Is Adaptive Green Hosting Useful when Services by Offsets Directly?** Instead of using green hosts, services could buy offsets directly, removing the need to route requests across multiple datacenters. As discussed in Section 3, services that adopt this approach lose economic benefits from bundling hosting and offsetting costs. Nonetheless, we can compute the carbon-offset elasticity for these services by treating carbon markets as a special Web host that offers many offsets and zero throughput. We divided $cost_{co2e}$ by the price of an EC2 instance and used the result (approx. 8000%) as the offset ratio for the special, carbon-market host. We added this host as a fourth choice to every service in our setup. Some services used this host, reducing the profit per service for the shared hosts. However, as shown in Figure 11, our adaptive approach still provided the most profit for shared green hosts, increasing profit by at least 7% compared to the over offsetting approach.

# 6. CONCLUSION

Green hosts invest in clean energy while keeping their prices low and competitive. These hosts profit from their investment by hosting more Internet services than their traditional counterparts; it is possible that they can tap into a niche market to accomplish this. Today's green hosts adopt ad-hoc policies for investing in clean energy, e.g., by buying as much clean energy as possible within a fixed budget. This paper showed that such fixed policies yield below optimal profit when the hosted Internet services support diurnal and bursty workloads and when the hosted services have diverse carbon footprint goals. We proposed a new research agenda: *adaptive green hosting*, where hosts invest in clean energy based on prior or predicted yield. We proposed a first-cut reactive solution that exploits heavier-than-exponential tails in Internet service workloads. Our reactive approach improves profit for existing green hosts and tends to urge hosts to increase their investments in clean energy. Future work will improve upon our approach by considering more complex interplay between SLAs and carbon footprint goals, heterogeneous energy efficiency and carbon efficiency among hosts, and in depth workload prediction approaches.

# 7. REFERENCES


[1] Amazon elastic compute cloud. `http://aws.amazon.com/ec2/pricing/`.

[2] Ecosia - number of daily searches. `http://ecosia.org/statistics.php`.

[3] T. Abdelzaher, J. Stankovic, C. Lu, R. Zhang, and Y. Lu. Feedback performance control in software services. *Control Systems, IEEE*, 23(3), 2003.

[4] Aiso.net: Web hosting as nature intended. http://aiso.net.

[5] M. Arlitt and C. Williamson. Web server workload characterization: The search for invariants. In *SIGMETRICS*.

[6] L. Barroso and U. Holzle. *The Datacenter as a Computer – An Introduction to to the Design of Warhouse-Scale Machines*. Morgan and Claypool Publishers, 2009.

[7] A. Bestavros and M. Crovella. Self-similarity in world wide web traffic: Evidence and possible causes. In *Sigmetrics*, 1995.

[8] J. Chattaway. Rackspace green survey. In *Rackspace Hosting White Paper*, June 2008.

[9] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *USENIX NSDI*, Apr. 2008.

[10] N. Deng and C. Stewart. Concentrating renewable energy in grid-tied datacenters. In *International Symposium on Sustainable Systems and Technology (ISSST)*, May 2011.

[11] N. Deng, C. Stewart, D. Gmach, and M. Arlitt. Policy and mechanism for carbon-aware cloud applications. In *NOMS*, Apr. 2012.

[12] Domain Tools.

[13] Ecosia - the green search. `http://www.ecosia.org/`.

[14] Eileen, OWC.net. Personal communication. http://owc.net.

[15] A. Gandhi, Y. Chen, D. Gmach, M. Arlitt, and M. Marwah. Minimizing data center sla violations and power consumption via hybrid resource provisioning. In *IGCC*, 2011.

[16] D. Gmach, Y. Chen, A. Shah, J. Rolia, C. Bash, T. Christian, and R. Sharma. Profiling sustainability of data centers. In *ISSST*, 2010.

[17] 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.

[18] GoodEnergy. `http://www.goodenergy.com`, 2012.

[19] Greenqloud- the worlds first truly green compute cloud. http://www.greenqloud.com.

[20] V. Griskevicius, J. M. Tybur, and B. Van den Bergh. Going green to be seen: status, reputation, and conspicuous conservation. *J Pers Soc Psychol*, 98(3):392–404, 2010.

[21] N. Holdings. The nielsen global online environmental survey, 2011.

[22] Hostgator: Web hosting services. http://hostgator.com.

[23] P. Inc. Going green:sustainable growth strategies. `www.pwc.com/en_GX/gx/technology/pdf/going-green.pdf`, 2011.

[24] ipath global carbon etn. `http://www.ipathetn.com/product/GRN/`.

[25] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley and Sons, 1991.

[26] James Packard, Emerson Inc. Personal communication. http://emerson-datacenter.com.

[27] J. Kaplan, W. Forrest, and N. Kindler. Revolutionizing data center energy efficiency. McKinsey and Company, 2008.

[28] 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.

[29] C. Li, W. Zhang, C. Cho, and T. Li. Solarcore: Solar energy driven multi-core architecture power management. In *HPCA*, Feb. 2011.

[30] K. Lim, P. Ranganathan, J. Chang, C. Patel, T. Mudge, and S. Reinhardt. Understanding and designing new server architectures for emerging warehouse-computing environments. In *ISCA*, 2008.

[31] Z. Liu, M. Lin, A. Wierman, S. Low, and L. Andrew. Geographical load balancing with renewables. In *Greenmetrics*, June 2011.

[32] Z. Liu, M. Lin, A. Wierman, S. Low, and L. Andrew. Greening geographical load balancing. In *ACM SIGMETRICS*, June 2011.

[33] LPSolve. `http://www.lpsolve.com`.

[34] D. Meisner, B. Gold, and T. Wenisch. Powernap: Eliminating server idle power. In *ASPLOS*, Mar. 2009.

[35] M. Ontkush. Plethora of options for green web hosting. `www.treehugger.com`, 2007.

[36] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu. No power struggles: Coordinated multi-level power management for the data center. In *ASPLOS*, Mar. 2008.

[37] P. Ranganathan and P. Leech. Simulating complex enterprise workloads using utilization traces. In *ASPLOS*, Mar. 2007.

[38] C. Ren, D. Wang, B. Urgaonkar, and A. Sivasubramaniam. Carbon-aware energy capacity planning for datacenters. In *MASCOTS*, Aug. 2012.

[39] N. Sharma, S. Barker, D. Irwin, and P. Shenoy. Blink: managing server clusters on intermittent power. In *ASPLOS*, Mar. 2011.

[40] C. Stewart, T. Kelly, and A. Zhang. Exploiting nonstationarity for performance prediction. In *EuroSys Conf.*, Mar. 2007.

[41] C. Stewart and J. Li. Power provisioning for diverse datacenter workloads. In *Workshop on Energy Efficient Design*, June 2011.

[42] 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.

[43] M. J. Todd. The many facets of linear programming. *Mathematical Programming*, 91(3), Feb. 2002.

[44] Trefis Team. Amazon kills it in cloud computing but it wont budge the stock price.

[45] The AMD Opteron Processor Helps AISO. www.vmware.com.

[46] Y. Zhang, Y. Wang, and X. Wang. Greenware: Greening cloud-scale data centers to maximize the use of renewable energy. In *Middleware*, Dec. 2011.