

# Stratus: Load Balancing the Cloud for Carbon Emissions Control

Joseph Doyle, Robert Shorten, and Donal O'Mahony

**Abstract**—Large public cloud infrastructure can utilize power which is generated by a multiplicity of power plants. The cost of electricity will vary among the power plants and each will emit different amounts of carbon for a given amount of energy generated. This infrastructure services traffic that can come from anywhere on the planet. It is desirable, for latency purposes, to route the traffic to the data center that is closest in terms of geographical distance, costs the least to power, and emits the smallest amount of carbon for a given request. It is not always possible to achieve all of these goals, so we model both the networking and computational components of the infrastructure as a graph and propose the Stratus system which utilizes Voronoi partitions to determine which data center requests should be routed to based on the relative priorities of the cloud operator.

**Index Terms**—Voronoi partitions, cloud computing, load balancing, carbon emissions



## 1 INTRODUCTION

A variety of new services are being offered under the cloud-computing paradigm. This service model involves a cloud-based service provider (CBSP) providing a large pool of computational and network resources which are allocated on demand to the cloud users from the pool. Cloud users in turn can use these resources to provide services for the users. The pool of resources can comprise of several data centers (DCs) at different geographical locations. There are many potential benefits to a global distribution of servers if load balancing is used correctly. Reduced latency and increased data transmission rates can be achieved by assigning clients to servers which are closer in terms of link distance. For some applications such as conference Voice-over-IP (VoIP) software and interactive online games, low latency is critical to provide a satisfactory quality of service (QoS). In addition, there have been proposals to consider electricity price when load balancing [1], [2], [3] to reduce operational costs. By assigning more load to a DC which is utilizing relatively cheap electricity operational costs can be lowered. This load balancing can be achieved with protocol-level mechanisms which are in use today such as dynamically generated DNS responses, HTTP redirection, and the forwarding of HTTP requests. All of these have been evaluated thoroughly [4], [5], [6].

Recently the carbon emissions associated with powering DCs have become important. Greenpeace report [7] the carbon emissions of selected DCs and the percentage of their electricity generated by power plants that use fuels which emit a relatively large amount of carbon. The carbon intensity of a power plant is the carbon emitted for a given

amount of energy generated. The carbon intensity of power plants using particular fuels is detailed in [8], [9]. Currently, there is little financial motivation to use green or clean energy, but increasing regulation of carbon emissions and schemes like the European Union Emissions Trading Scheme (EU ETS) [10] mean that in the future it is probable that the right to emit carbon into the atmosphere will be traded as a commodity. In addition, recent work [11] suggests that on-site power generation can reduce carbon emissions and electricity cost by reducing the peak draw of a data center from an electricity supplier.

There have been some proposals to use locally generated clean energy [12] or employ load balancing based upon the carbon intensity of the electricity supplier [13]. These proposals, however, do not consider the carbon emitted as a result of packets travelling across the network from the client to the server. While the energy consumed by the networking equipment as part of the cloud computing has been analyzed [14], additional analysis is required to examine the total carbon emission caused by a cloud-computing system.

In addition, other proposals for minimizing carbon emissions use weather data as a metric for load balancing. While this is a useful metric for in-house generated electricity, it can be inaccurate when electricity is obtained from an external supplier as other factors affect their carbon intensity. This is discussed in greater detail in Section 5.2. Carbon emissions are seldom the sole concern of cloud operators and other factors must be considered. The electricity cost can vary considerably between different geographical regions and this fact can be exploited by cloud operators to lower the operational cost.

The manner in which a data center is cooled can affect both the electricity cost and carbon emissions as certain schemes such as “free air cooling” require less energy, and hence emit less carbon. Finally, cloud operators are usually bound by service level agreement (SLA), and therefore must maintain a minimum QoS for service users.

It is not always possible to achieve the best case outcome for all of these factors as they sometimes conflict, so we

- J. Doyle and D. O'Mahony are with the CTVR Research Centre, Trinity College Dublin, Oriel House, Dublin 2, Ireland.
- R. Shorten is with the IBM Research Ireland, Ireland.

Manuscript received 29 Apr. 2013; revised 9 Aug. 2013; accepted 14 Aug. 2013; published online 23 Aug. 2013.

Recommended for acceptance by A. Wierman.

For information on obtaining reprints of this article, please send e-mail to: [tcc@computer.org](mailto:tcc@computer.org), and reference IEEECS Log Number TCC-2013-04-0081. Digital Object Identifier no. 10.1109/TCC.2013.4.

formulate a graph-based approach which we call Stratus that can be used to examine and control the operation of the cloud. Stratus uses Voronoi partitions which are a graph-based approach which have been used to solve similar problems in other areas such as robotics [15]. In this paper, we use this approach to attempt to control the factors which affect the operation of the cloud. This paper makes the following contributions:

- The development of a model detailing the carbon emissions, electricity cost, and time required for the computational and networking aspects of a service request.
- A distributed algorithm which minimizes the combination of average request time, electricity cost, and carbon emissions is described.
- Data for the carbon intensity and electricity price of various geographical regions and a representative set of round trip time between various geographical regions is presented.
- We evaluate the performance of our distributed algorithm using the data obtained for various scenarios.

## 2 RELATED WORK

There have been a number of proposals which consider the cost of electricity when determining which data center should service requests. Qureshi et al. [1] proposed a distance-constrained energy price optimizer and presented data on energy price fluctuations and simulations illustrating the potential economic gain. Stanojević et al. [2] detail a distributed consensus algorithm which equalizes the change in the cost of energy. This is equivalent to minimizing the cost of energy while maintaining QoS levels. Rao et al. [16] formulate the electricity cost of a cloud as a flow network and attempt to find the minimum cost of sending a certain amount of flow through this network. Rao et al. [17] also propose a control system which uses load balancing and server power control capabilities to minimize energy cost. Wang et al. [18] propose using a corrected marginal cost algorithm to minimize electricity cost. Mathew et al. [19] propose an algorithm which controls the number of servers online in the cloud to reduce energy consumption. It also maintains enough servers at each data center to handle current requests as well as spare capacity to handle spikes in traffic. Liu et al. [3] propose distributed algorithms, which minimize the sum of an energy cost and a delay cost using optimization techniques such as gradient projection, to minimize the overall cost of operating the data center. In addition, they expand their formulation to consider minimizing the sum of the social-impact cost and delay cost. They define the social-impact cost as a metric for environmental impact of the data center. By examining the availability of renewable energy and directing load to the appropriate data centers they attempt to reduce the environmental impact of the data center.

In addition, there has been some analysis of the electricity consumption of the cloud-computing paradigm. Baliga et al. [14] analyze the power consumption of all the elements of this for a variety of service scenarios. Mahdevan

et al. [20] examine the power consumption of network switches and consider techniques for improving the power efficiency of network switches by disabling ports and using lower data rates where possible.

There have also been some proposals which consider carbon emissions when determining where to direct service requests. Liu et al. [12] expand the model proposed in [3] to subtract locally generated clean energy from the energy cost calculation to allow data centers which have clean energy generation facilities to service more load. Doyle et al. [13] describe an algorithm that minimizes a cost function containing the carbon intensity of the electricity supplier of the data center and average job time. Moghaddam et al. [21] attempt to use a genetic algorithm-based method with virtual machine migration to lower the carbon footprint of the cloud. Gao et al. [22] use a flow-optimization-based framework to control the three-way tradeoff between average job time, electricity cost, and carbon emissions. This system, however, is only evaluated using yearly average carbon intensity values. These can differ significantly from the instantaneous carbon intensity value of an electricity supplier. An important distinction between this work and the work of Gao et al. is the use of Voronoi partitions to enable load balancing.

Some of these proposals use various mathematical techniques to achieve their goals. In this work, we propose the use of Voronoi partitions which are used in a number of areas. Aurenhammer details a number of applications in [23]. Durham et al. [15] use Voronoi partitions to divide an environment so that a group of robots can provide coverage.

This problem can be viewed as similar to a constrained version of the facility location problem which has been shown to be NP-hard [24]. Exact [25] and approximate [26] solvers for this problem have been thoroughly studied. These methods, however, are computationally expensive as any new demand points require the solver to run the entire analysis from scratch. This works well for determining the optimal site for the construction of a facility as the demand set is static. In the cloud, however, the demand for a service is constantly changing and a system that can respond to incremental change is required.

## 3 PROBLEM FORMULATION

In this section, we formulate the problem. To do this we need some background notation. Namely, we need to say what a graph is, what a Voronoi partition is, and how we use these ideas in the context of cloud computing.

### 3.1 Graph

A graph consists of a finite set of nodes and edges. Each edge is incident with two nodes. A path is an ordered sequence of points such that any consecutive pair of points is linked by an edge in the graph. In an undirected graph, there is no direction associated with the edges. Hence, a path can be constructed with any edge in the graph. A weighted graph associates a label with each edge. Nodes are connected if a path exists between them.

### 3.2 Voronoi Partitions

Voronoi partitions are the decomposition of a set of points into subsets. These subsets are centered around points

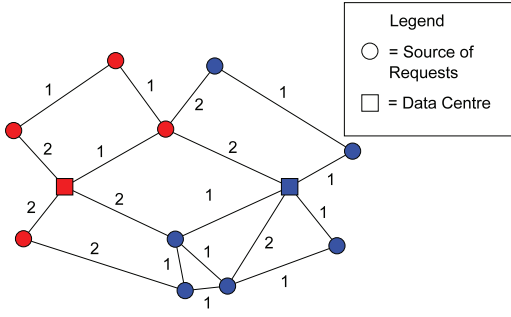


Fig. 1. Example of how sources of requests are partitioned between two data centers. Color indicates that the node is part of a particular partition.

known as sites, generators, or seeds. Each point in the set is added to a subset consisting of a site and all other points associated with this site. An abstract notion of distance between a point and the sites is used to determine which subset a point is associated with. A point is assigned to a subset if the distance to site is less than or equal to the distance to the other sites. For an example of Voronoi partitions used in applications (robotics) (see [15]). We shall now use these partitions to solve routing problem associated with load balancing in the cloud.

### 3.3 Voronoi Partitions of the Cloud

In our work, the set of points consist of sources of requests for cloud services and data centers which service these. Voronoi partitions are then used to determine where requests are serviced. A Voronoi cell represents which sources of requests a data center is servicing at a given time. An example of a group of sources of requests which have been partitioned between two data centers can be seen in Fig. 1. In this figure each source of requests has a path to both data centers. The partition that the source of requests is a part of depends on the paths to the two data centers. The partitions are made up of sources of requests which have paths available to them with lower distances than the paths available to the other data center.

### 3.4 Problem Statement

Let  $|J|$  be a set of  $J$  geographically concentrated sources of requests and  $|N|$  be a set of  $N$  data centers. Let  $|Q|$  be a finite set of points that represent either sources of requests or data centers. These points are connected by  $E$  edges in an undirected weighted graph  $\mathcal{G} = (|Q|, |E|, |w|)$ . The weights are calculated as functions of the time required to service a fraction of the request  $T_i$ , the carbon emissions associated with servicing the fraction  $G_i$  and the electricity cost  $E_i$ , if any associated with servicing the request along the edge.

$$w_i = f(T_i, G_i, E_i) = T_i + R_1(G_i) + R_2(E_i) \quad \forall i \in |w|,$$

where  $R_1$  and  $R_2$  are the relative price functions which are used to specify the relative importance of the factors. While  $E_i$  and  $G_i$  are related, the rates at which they increase may vary significantly depending on the specifics of the cloud and hence, both must be included in the problem formation to ensure that the cloud operator can operate the cloud as desired. It should be noted that the weights of the graph represent the networking and computational aspects of servicing a request.

```

1   $U := P_i(t) \cup P_j(t)$ 
2  for  $x \in U$ 
3       $W_i := \{x \in U : d(x, i) \leq d(x, j)\}$ 
       $W_j := \{x \in U : d(x, i) > d(x, j)\}$ 
4  endfor
5   $P_i(t+1) := W_i$ 
    $P_j(t+1) := W_j$ 

```

Fig. 2. Pseudocode for pairwise partitioning rule.

The set  $|Q|$  is partitioned into  $N$  subsets representing the regions serviced by each data center. This results in a collection  $P = \{P_i\}_{i=1}^N$  of  $N$  subsets of  $|Q|$  such that:

1.  $\bigcup_{i=1}^N P_i = Q$
2.  $P_i \cap P_j = \emptyset$  if  $i \neq j$
3.  $P_i \neq \emptyset \quad \forall i \in \{1, \dots, N\}$
4.  $P_i$  is connected for all  $i \in \{1, \dots, N\}$

Two subgraphs  $P_i$  and  $P_j$  are connected if there are two vertices  $q_i$  and  $q_j$  belonging, respectively, to  $P_i$  and  $P_j$  such that  $(q_i, q_j) \in |E|$ .

We can use Voronoi partitions to establish a collection of subsets which minimizes the combination of carbon emissions, electricity cost, and average request time. In this case, the Voronoi partition  $P_i$  associated with data center  $i \in |N|$  can be defined as the set of points whose distance to data center  $i$  is less than or equal to the distance to another data center  $j \in |N|$ . We assume that the data centers have sufficient computational capacity so that there is no constraint on the size of  $P_i$ . To compute this we need to define how the distance between two points is calculated. A standard notion of distance between two points  $d(i, j)$  in a weighted graph is the lowest weight of a path between the two points  $(i, j)$ . The weight of a path is the sum of the weights of the edges in the path. The goal of using the Voronoi partitions in this scenario is to minimize the distance between the sources of requests and the data centers. This can be defined as

$$\min \sum_{i=1}^N \sum_{j \in P_i} d(i, j).$$

Note, if a source is equidistant to more than one data center the point is assigned to the Voronoi partition that has the least members to attempt to balance the load on the data centers.

A pairwise partitioning rule can be used to achieve this goal.

## 4 PAIRWISE PARTITIONING RULE

At time  $t$  data center  $i$  and data center  $j$  communicate by exchanging the partitions  $P_i$  and  $P_j$ , so that each data center can examine all the regions associated with the two data centers to determine if there is a better route available between a data center and a region. We assume without a loss of generality that  $i < j$ . Each data center then performs the actions depicted in the pseudocode in Fig. 2. The paths between each region and the two data centers are examined. If the path between the data center  $i$  and a region is smaller than the path between the region and the

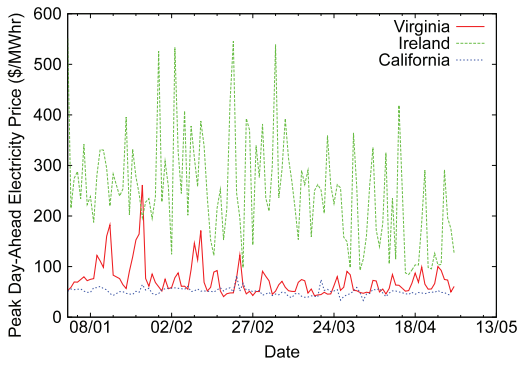


Fig. 3. Peak daily price of electricity for suppliers in the regions of the three data centers studied.

data center  $j$  then the region is added to a temporary partition associated with data center  $i$ . Otherwise, it is added to a temporary partition associated with data center  $j$ . The partitions of the two regions are then updated with the appropriate temporary partition. To generate the initial partitions the distance between each node in the graph and all the data centers is calculated. The nodes are then added to the partition which yields the minimal distance between the nodes and the data center.

## 5 CLOUD ANALYSIS

In this section, we examine the variation in the costs that exist between data centers.

### 5.1 Electricity Cost

The price of electricity on the wholesale market depends on a number of factors. The wholesale electricity market is administered by an authority known as a regional transmission organization (RTO) in the United States and the single electricity market operator (SEMO) in Ireland. In this market, power producers present supply offers, consumers present bids, and an authority in charge of the transmission network determines how the electricity should flow and sets prices [1]. The price is determined based on the bids and offers as well as other factors such as reliability and grid connectivity. The variation of local electricity prices in different geographical regions can be exploited by cloud operators to lower operational costs [1]. To illustrate this, we examine the potential savings that can be made by part of Amazon's EC2 [27] cloud. We examine the local prices of electricity suppliers located in the regions of the California, Virginia, and Ireland data centers. Pacific Gas and Electric (PG&E) is one supplier in the California region and Dominion (DOM) is a supplier in the Virginia region. We chose these as they supply electricity in the region where the data centers are located. Ireland uses a single market for electricity known as SEMO and only a single price for wholesale electricity is available. The peak daily day-ahead electricity price for these suppliers from January 2011 through April 2011 is depicted in Fig. 3.

It is interesting to note that the maximum price can approach \$550/MWh and that the peak price for the electricity is nearly always greatest in the Ireland region. This would suggest that little traffic would be routed to the Ireland data center if a load-balancing scheme design to

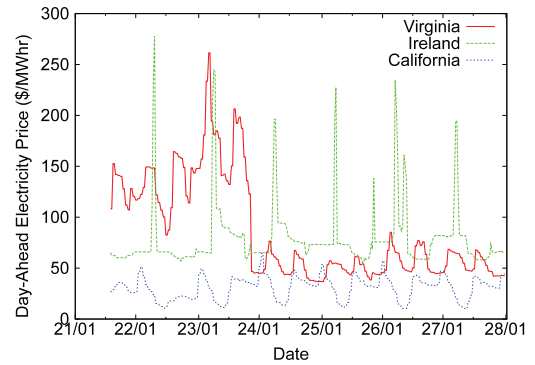


Fig. 4. Price of electricity for suppliers in the regions of the three data centers studied.

minimize electricity prices was utilized. If, however, we examine the hourly variations of electricity prices we can see that this is not the case. The day-ahead electricity price for the electricity suppliers from the 22nd January 2011 through the 29th January 2011 is depicted in Fig. 4. From this we can see that peaks in electricity price in the Ireland region tend to be very sharp and that at nonpeak times the variation in price between geographical regions is much smaller.

### 5.2 Carbon Emissions

An analysis of the carbon intensity of electricity suppliers in various geographical regions is useful when attempting to minimize the environmental impact of a cloud. To illustrate this, we examine the carbon emitted by a service which has users in a number of different geographical regions utilizing the EC2 infrastructure. The carbon intensity data for the data centers and sources of requests were obtained from [28] and can be seen in Table 1. The data for states in the United States were in agreement with data from [29]. The carbon intensity of an electricity supplier is calculated using the weighted average (where the power generated by the power plant is the weight used) of the carbon intensity of the power plants operated by the electricity supplier. The demand for electricity changes over the course of a day and electricity suppliers turn power plants on and off to react to the changes in the demand. A consequence of this is that the carbon intensity of an electricity supplier varies over time. It would be possible to estimate the realtime carbon intensity by examining the weighted average of the carbon intensity for all the power plants that are operating, but some electricity suppliers provide a realtime carbon intensity value directly. To the author's knowledge, the realtime carbon intensity of all the geographical regions in Table 1 is not available. It is, however, available for the Ireland region. Fig. 5 depicts the daily peak carbon intensity of the electricity supplier in Ireland from January through April 2011. This data was obtained from the Ireland Transmission System Operator Eirgrid [30]. We can see that there is a large variation with time. This suggests that the data can be exploited to minimize the environmental impact of the cloud. Fig. 6 depicts the carbon intensity of the SEMO suppliers from the 22nd January 2011 through the 29th January 2011. The interval between data points is fifteen minutes. From Figs. 4 and 6 we can see that the carbon intensity is not as volatile as the electricity market price but

TABLE 1  
Average Round Trip Time between Data Centers and Sources of Requests, Carbon Intensity of Data Centers and Sources of Requests, and Daily Number of Requests at Source

Region	California (ms)	Ireland (ms)	Virginia (ms)	Carbon Intensity (g/kWhr)	Number of Requests (Millions)
Austria (AUS)	177.98	47.67	159.07	870	7.038
Belgium (BEL)	171.98	28.45	158.09	317	11.736
California (CAL)				384	
Colorado (COL)	42.77	155.36	101.27	903	6.76
Connecticut (CON)	88.05	117.65	75.73	392	4.293
Finland (FIN)	188.47	55.77	176.72	99	5.418
Florida (FLO)	54.26	171.87	98.21	762	26.365
France (FRA)	192.44	21.24	184.75	96	61.355
Georgia (GEO)	58.91	115.12	77.43	694	1.968
Germany (GER)	177.74	40.89	157.68	612	58.76
Illinois (ILL)	63.81	142.78	102.08	544	18.049
Indiana (IND)	69.65	151.05	83	986	7.803
Ireland (IRE)				655	
Italy (ITA)	188.71	44.71	167.3	473	55.372
Kansas (KAN)	50.48	148.4	85.2	817	4.545
Kentucky (KEN)	71.98	146.85	87.29	968	5.169
Maryland (MAR)	99.71	140.46	88.05	641	6.69
Massachusetts (MAS)	89.33	98.02	72.1	603	9.602
Minnesota (MIN)	62.09	147.74	85.79	744	6.724
Netherlands (NET)	163.93	19.71	138.79	548	15.527
New York (NEW)	96.45	78.71	134.11	386	27.604
North Carolina (NCA)	72.32	72.45	31.12	604	11.817
Norway (NOR)	194.82	48.84	183.08	6	6.69
Ohio (OHI)	83.81	132.08	69.17	873	14.828
Oklahoma (OKL)	46.42	159.96	98.22	819	4.378
Ontario (ONT)	90.84	142.81	97.12	224	16.64
Oregon (ORE)	27.66	213.48	153.28	246	4.807
Pennsylvania (PEN)	71.99	118.53	52.78	597	16.097
Portugal (POR)	222.69	64.11	190.02	550	10.925
Spain (SPA)	194.55	35.83	172.47	487	40.633
Sweden (SWE)	186.01	48.79	170.28	19	11.887
Tennessee (TEN)	235.61	276.43	311.61	661	7.891
Texas (TEX)	37.61	151.93	98.96	763	31.015
UK (UK)	175.59	17.62	163.25	614	78.647
Virginia (VIR)				559	
Washington (WAS)	29.57	192.11	126.53	938	10.119
Wisconsin (WIS)	67	146.49	94.03	834	7.025

varies enough to allow the cloud operator to utilize the realtime data to minimize the environmental impact.

A novel aspect of our approach to minimizing carbon emissions when compared with other approaches [3], [21] is that we use carbon intensity data rather than weather data when determining where to route load. This does not affect schemes where power is generated locally by the cloud operator, but it can have a significant effect when cloud

operators draw power from an electricity supplier for two reasons. First it is not always possible to utilize solar and wind power. An electricity network must carefully balance supply and demand and ideally the market authority would use the cleanest power plants available to meet the demand. This, however, cannot be achieved in reality as it would require power plants to be able to turn on or off in very short spaces of time and some power plants (e.g., coal)

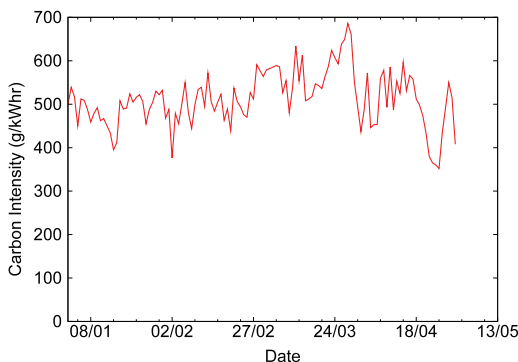


Fig. 5. Daily peak carbon intensity of electricity supplier in the region of the Ireland data center studied.

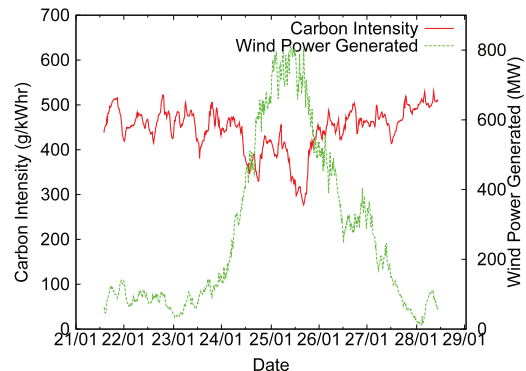


Fig. 6. Carbon intensity and generated wind power of electricity supplier in the region of the Ireland data center studied.



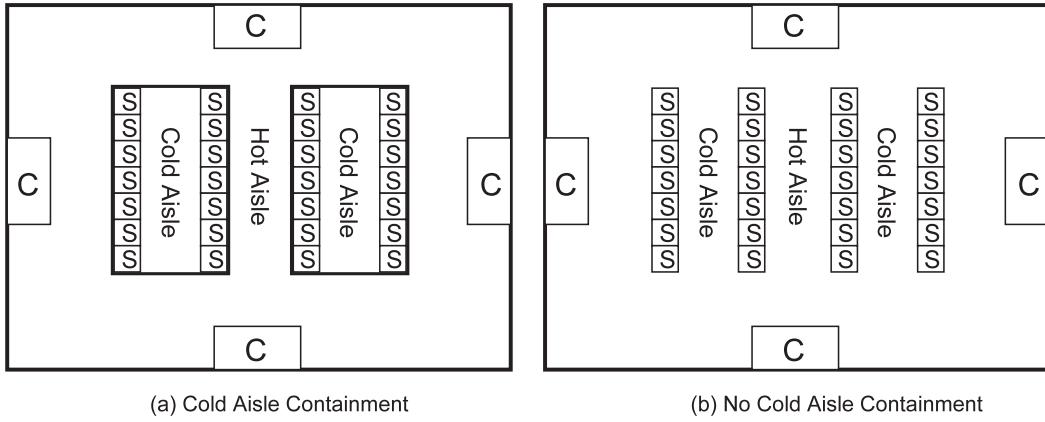


Fig. 7. Layout of cooling cost simulations with (a) cold aisle containment and (b) no cold aisle containment.

take a long time to turn on or off. The result of this is that they are very rarely turned off and if there is insufficient system demand, solar and wind power is wasted.

The second reason that the use of weather data can be an inaccurate metric is that even if there is sufficient demand and solar and wind power is utilized, the changes in the operation of other power plants can affect the carbon intensity. As a result, there is not a direct correlation between availability of wind and solar power and carbon emissions. For example, if a pumped storage plant is turned on and the wind speed drops, carbon intensity may still go down. The reason for this is that the reduction in carbon emissions caused by the use of the pumped storage plant may be greater than the increase in carbon emissions caused by other power plants supplying the electricity which is no longer supplied by the wind turbines. If we examine Fig. 6 we can see an example of this. There is some correlation between the wind power generated and carbon intensity, but it is not direct. Sometimes when the wind power generated increases the carbon intensity also increases. It should be noted that the Irish SEMO market does not use significant amounts of solar power so this is not a factor in the analysis.

### 5.3 Cooling Cost

Cooling costs for a data center are dependent on its design and the local climate in addition to the load placed upon it. If a data center uses aisle containment [31] it can significantly reduce the cost of cooling the data center. Aisle containment is the separation of the inlets and outlets of servers with a barrier such as PVC curtains or Plexiglas [32] to prevent air migration, which adversely affects cooling costs.

In addition “free air cooling” can be used. This is the use of air economizers to draw in cold air from the environment into the data center when the climate conditions are suitable, thereby, preventing the use of computer room air conditioner (CRAC) chiller units and lowering the cooling costs [33]. Water cooling [34], [35] can also be used, but it is rarely used in data centers at present.

To examine how this cost varies with demand, we constructed two models of data centers in the computational fluid dynamics (CFD) simulation software Flovent [36]. These represent typical data centers which have been

examined in previous research [37], [38]. One data center used cold aisle containment and the other did not. Apart from this the data centers were of similar construction. Each data center has dimensions  $11.7 \text{ m} \times 8.5 \text{ m} \times 3.1 \text{ m}$  with a  $0.6 \text{ m}$  raised floor plenum that supplies cool air through perforated floor tiles. There are four rows of servers with seven 40U racks in each case, resulting in a total of 1,120 servers. The servers simulated were based on a Hewlett-Packard’s *Proliant DL360 G3s* model, which consumes 150W of power when idle and 285W at 100 percent utilization. From this we can determine that the total power consumption of the data center is 168kW when idle and 319.2kW at full utilization. For cooling, the data center is equipped with four CRAC units. Each CRAC unit pushes air chilled to  $15^\circ\text{C}$  into the plenum at a rate of  $16,990 \text{ m}^3/\text{h}$ . The cooling capacity of each CRAC unit is limited to 90kW, and in full operation each CRAC unit itself consumes 10kW.

The layout of the two data centers modeled is shown in Fig. 7. Racks of servers are represented as boxes with the letter “S” and CRAC units can be identified as boxes with the letter “C”. The simulations are used to establish the maximum inlet temperature of a server rack  $T_{max}$ . We can use this to establish the cooling costs  $C$  which can be calculated as follows:

$$C = \frac{Q}{\text{COP}(T_{sup} + (T_{safe} - T_{max}))} + P_{fan}, \quad (1)$$

where  $Q$  is the amount of power the servers consume,  $T_{sup}$  the temperature of the air that the CRAC units supply,  $T_{safe}$  the maximum permissible temperature at the server inlets to prevent equipment damage,  $T_{max}$  the maximum temperature of the server inlets in the data center,  $P_{fan}$  the power required by the fans of the CRAC units, and COP is the “coefficient of performance” (COP), that is the ratio of heat removed to work necessary to remove the heat, is a function of the temperature of the air being supplied by the CRAC unit. The COP of a typical chilled-water CRAC unit used in the calculations of cooling costs is depicted in Fig. 8. We assume a  $T_{safe}$  value of  $25^\circ\text{C}$ .

Fig. 9 depicts the results of the cooling-cost simulations. The percentage utilization of the data center is shown on the  $x$ -axis and the cooling cost in watts is shown on the  $y$ -axis.. Each line represents a different system. The CAC-CRAC line is a system which uses cold aisle containment and CRAC

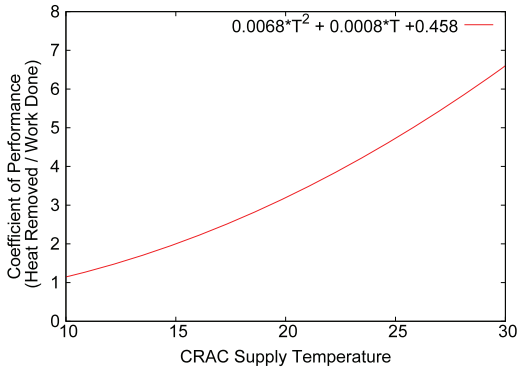


Fig. 8. Typical model of the Coefficient of Performance (COP) curve for a chilled water CRAC unit.

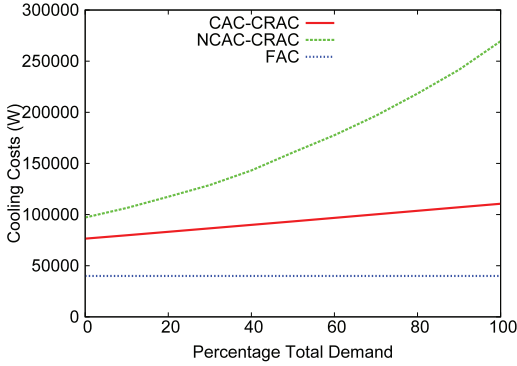


Fig. 9. Cooling cost of various data center cooling systems at various levels of demand.

cooling. The FAC line is a system uses “free air cooling”. The NCAC-CRAC line is a system which does not use cold aisle containment and CRAC cooling. “Free air cooling” only consumes fan power, and is therefore constant.

#### 5.4 Average Job Time

The previous sections establish that electricity cost and carbon emissions can be lowered. It is likely, however, that there will be an increase in the average service request time which the cloud operator will have to take into account when determining its load-balancing policy. A useful metric for service request time is the latency between the server and client. To establish the round trip time data an experiment on PlanetLab [39] was established with a server at each node location. Nodes in the same region as our three data center locations then pinged the other geographical regions at 15 minute intervals for approximately two days. The average latency established from this experiment can be seen in Table 1. Average service request time could be reduced by routing load from a geographical region to the data center region using lowest latency as a criterion to route load. From Table 1 we can see that if such a load-balancing scheme was used, each data center region will have some of the load routed to it. From this, we can conclude that any reduction in carbon emissions or electricity cost will cause an increase in the average latency as load will not be routed with latency as the sole metric.

Figs. 10, 11, and 12 depict the measured latency between the data center and the other regions for the California, Ireland, and Virginia data centers, respectively. It is interesting to note that the latencies remain mostly constant

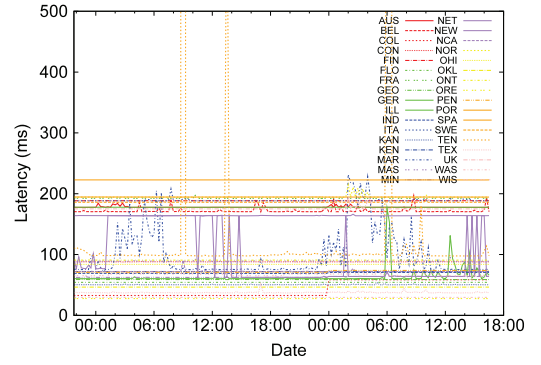


Fig. 10. Latency between California and different geographical regions at fifteen minute intervals.

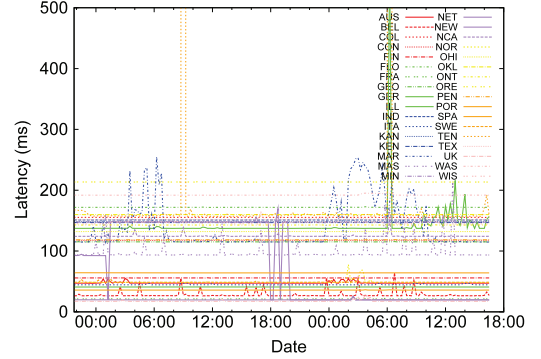


Fig. 11. Latency between Ireland and different geographical regions at fifteen minute intervals.

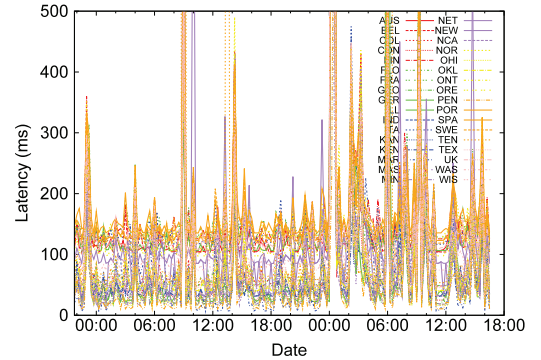


Fig. 12. Latency between Virginia and different geographical regions at fifteen minute intervals.

over time at the California and Ireland data centers, but vary frequently at the Virginia data center. We postulate that this is a result of congestion at the Virginia region. In addition, we can see that there is some variation in the latency between the other regions and the California and Ireland data centers. Thus, we can conclude that latency varies with time particularly in regions where congestion takes place and it should be monitored so that the increase in average service request time caused by reducing carbon emissions or electricity cost can be measured correctly. Indeed, there may be times, when there is no increase in average service request time associated with a reduction in carbon emissions or electricity cost.

## 6 SIMULATION SETUP

In this section, we describe the setup for the simulation of the algorithm described in Section 4 and our methodology

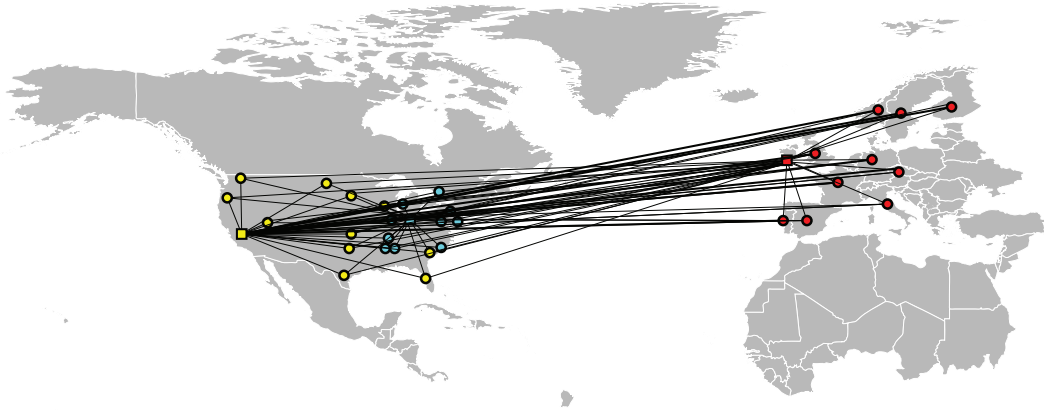


Fig. 13. Diagram of the simulation setup. The color of the node indicates that the node is part of a particular partition.

for establishing the weights of the graph described in Section 3.4. We simulate three data centers. One of these is in Ireland and the other two are in the United States in Virginia and California. We chose these locations to mimic Amazon’s EC2 platform [27] which currently has major data centers at these locations. We model 34 sources of requests in the simulation which represent certain countries in Europe, states in the United States, and provinces in Canada. Each source is connected to each data center by a single edge. This is illustrated in Fig. 13.

To calculate the weights of the edges of the graph we needed to determine the time, carbon emissions, and electricity cost associated with servicing the networking and computational portions of a request. For the networking portion of a request, we first assume that each service request requires the transfer of relatively small amount of data, and therefore the duration of the connection can be approximated by the round trip time. To calculate the time associated with the computational portion of the request we assume that the request requires 50 ms of computation.

To calculate the carbon emissions and electricity cost of serving a request we assume that the data center uses Hewlett-Packard’s *Proliant DL360 G3s*. This type of server consumes 150 W at 0 percent utilization and 285 W at 100 percent utilization. This yields dynamic power of 135 W for each server. This is then multiplied by the time required to service the computational portion of the request (50 ms) to yield the energy required to service a request. We must then consider the energy required for the additional cooling required by servicing the requests. We assume that the Ireland data center uses “free air cooling”, the Virginia data center uses cold aisle containment, and the California data center uses a standard cooling system with no cold aisle containment. The cooling energy required by the data center when it is not processing the request is subtracted from the energy required when it is processing the request to give the cooling energy caused by the request. This is added to the energy already calculated to yield the total computational energy. The total computational energy is then multiplied by the electricity price to yield the electricity cost  $E_i$ . The energy is also multiplied by the carbon intensity of the data center to yield the computational carbon emitted.

The networking aspect of the weights must also be considered. The power consumed by a switch can be altered

by powering off (disabling) ports when they are not in use and powering on (enabling) ports when they need to be used. To calculate the carbon emissions associated with servicing the networking portion of the request, we assumed that only two ports would open during the duration of the request. To calculate the carbon emitted, we first obtain the energy consumed by multiplying the duration of the round trip by a power value required to open a port (0.7 W). This value was a midrange value from those presented in [20]. The energy consumed is then multiplied by the average carbon intensity of the source of the request and the data center to obtain the network carbon emitted. This is added to the computational carbon emitted to give the total carbon emitted and the carbon weight  $G_i$ . While cloud operators are likely to be held at least partially responsible for the carbon emissions of the network by increased regulation, current modes of operation suggest that they are not held responsible for the vast majority of the electricity cost associated with networking and as such it is ignored.

In all simulations the algorithm runs for a two-day period using latency, electricity price, and carbon intensity data described in Section 5. The latency between the sources of requests and the data center is as seen in Figs. 10, 11, and 12. The electricity price is as seen in the first two days of Fig. 4 for the three data centers. The carbon intensity data is as seen in Table 1 for all the regions except Ireland which uses the data seen in the first two days of Fig. 6. The algorithm updates every 15 minutes. It is assumed that the time required to service a request is sufficiently short, and that redirection is not required to minimize the cost when the algorithm updates. It should be noted that we assume that redirecting additional requests to each data center does not cause congestion or affect the average service request time.

To examine the overall costs to the cloud, we must consider the number of requests coming from each source of requests. We used figures from the websites [40], [41] which estimated the number of Facebook users in each source location and assumed that the daily average number of service requests from a single user was 2.6. We used Facebook as it is representative of a broad range of cloud applications. The daily number of requests for each source can be seen in Table 1. We also needed to establish the number of requests at each source during each 15 minute interval over the two-day period. It has previously been



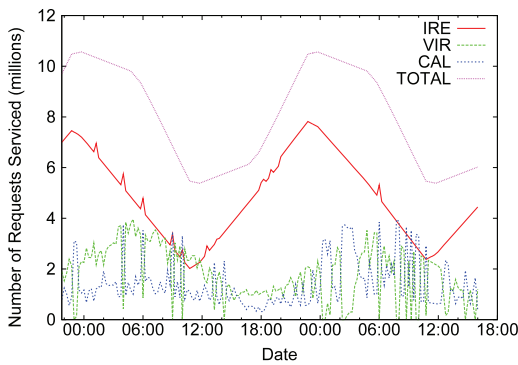


Fig. 14. Number of requests serviced at each data center when the “Best Effort Time” scenario is used. The time zone of the  $x$ -axis is GMT.

found that realistic workloads have a diurnal cycle with a trough at approximately 6:00 am and a peak of roughly four times the trough value at approximately midnight [1]. We divided the daily number of requests at each source into this diurnal pattern and adjusted the peaks to match the time difference of the region. The total demand can be seen in Fig. 14.

In the first set of simulation, we examine the extremes of the algorithm by looking at four scenarios. In the first scenario, we set the relative price functions to zero. This represents a scenario where time is crucial and the operator is attempting to minimize the time taken to service a request with no regard for electricity cost and associated carbon emissions  $R_1(G_i) = 0, R_2(E_i) = 0$  and the weights of the edges of the graph become  $w_i = T_i$ . We shall hereafter refer to this scenario as “Best Effort Time.” In the second scenario, we set the first relative price function to 10,000 times the carbon emissions. This essentially functions as infinite times the carbon emission. This represents a scenario where time and electricity cost are unimportant and all efforts can be made to reduce the associated carbon emissions  $R_1(G_i) = 10000G_i, R_2(E_i) = 0$  and the weights of the edges of the graph become  $w_i = T_i + 10000G_i$ . We shall hereafter refer to this scenario as “Best Effort Carbon.” In the third scenario, we set the second relative price function to 10,000 times the electricity cost. This represents a scenario where time and carbon emissions are unimportant and all efforts can be made to lower the electricity costs  $R_1(G_i) = 0, R_2(E_i) = 10000E_i$  and the weights of the edges of the graph become  $w_i = T_i + 10000E_i$ . We shall hereafter refer to this scenario as “Best Effort Electricity.” In the final scenario, we examine

a baseline for current load-balancing operations by examining a roundrobin scheme. This is the default option in many commercial load-balancing solutions. We shall hereafter to refer to this scenario as “RoundRobin”.

In the second set of simulations, we explore scenarios where the cloud operator needs to strike a balance between the three factors. In this set of simulations we examine the performance of the algorithm under scenarios which attempt to balance the various factors by adjusting  $(\alpha, \beta)R_1(G_i) = \alpha G_i, R_2(E_i) = \beta E_i$  in intervals of 100 from 0 to 10,000 and examining the total electricity cost, carbon emissions, and average service request time of each scenario to examine what savings in electricity cost and carbon emissions can be made when there are constraints on the average service request time. We define  $\alpha$  as a variable which represents the relative importance of carbon emissions to the cloud operator and  $\beta$  as a variable which represents the relative importance of electricity cost.

## 7 RESULTS

The key performance metrics for the simulations are the average service request time, the electricity cost, and the carbon emissions associated with servicing requests for the two days. We first examine these for the first set of simulations. They are shown for each scenario in Table 2. When comparing the best-effort carbon scenario with the roundrobin baseline we can see that carbon emissions for a service can be reduced by 21 percent. If we examine the best-effort electricity scenario and the roundrobin baseline we can see that the electricity cost can be reduced by 61 percent. There is, however, a corresponding increase in the average service request time of 7 ms. If we investigate the best-effort time scenario and the roundrobin baseline we can see that the average service request time can be reduced by 47 percent. It is also interesting to compare the three best-effort scenarios. If we compare the best-effort time scenario and the best-effort carbon scenario we see that the latter emits 13 percent less carbon, but has an average service request time that is 42 ms higher. If we examine the best-effort time scenario and the best-effort electricity scenario we can see that the latter costs 58 percent less, but has an average service request time that is 87 ms higher. These comparisons are useful for the cloud operator as it allows them to see if the scenarios are feasible under SLAs and whether it is more desirable to concentrate on lower electricity costs or carbon emissions.

TABLE 2  
Average Service Request Time, Daily Carbon Emission and Number of Requests Serviced at Each DC for Various Scenarios

Scenario	Average Service Request Time (ms)	Carbon Emissions (kg)	Electricity Cost (\$)	Number of Requests Serviced by California (million)	Number of Requests Serviced by Ireland (million)	Number of Requests Serviced by Virginia (million)
Best Effort Time	90	1378	240	241	822	302
Best Effort Carbon	132	1200	195	0	1365	0
Best Effort Electricity	177	1567	100	1276	0	89
RoundRobin	170	1522	257	455	455	455

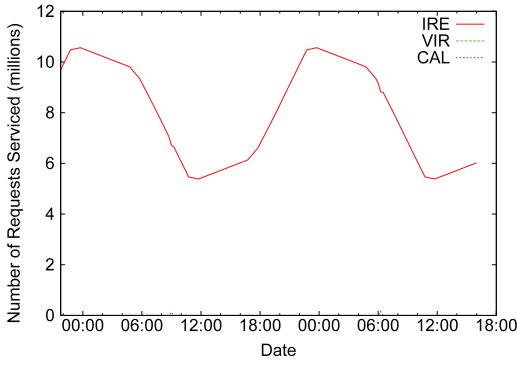


Fig. 15. Number of requests serviced at each data center when the “Best Effort Carbon” scenario is used. The time zone of the  $x$ -axis is GMT.

From Table 2 we can also see the number of requests sent to each data center under the various scenarios. We can see that under the best-effort carbon scenario all of the requests go to the Ireland data center. This is interesting as the carbon intensity for the California region is relatively low. The additional carbon caused by the cooling setup used in the data center is sufficiently high that all the requests go to the Ireland data center under this scenario. We can also see that under the best-effort electricity scenario most of the load goes to the California data center. It is interesting that the additional electricity cost of the cooling setup in that data center is mostly insufficient to overcome the local electricity price differential and the requests mostly go to the California data center. We also examine the number of requests serviced at the data centers at each time interval. Fig. 14 depicts the number of requests serviced at each data center during a time interval for the best-effort time scenario. From Fig. 14, we can see that the number of requests serviced at the Ireland data center follows the change in demand relatively steadily while the number of service requests for the Virginia and California data centers fluctuate. The reason for this is that the Ireland data center will take most requests from the European regions as the latency remains relatively steady, while the California data center is likely to take service requests which would otherwise go to the Virginia data center if there is congestion as the latency between these regions and the California data center is lower than the latency between these regions and the Ireland data center. Fig. 15 depicts the number of requests serviced at each data center during a time interval for the best-effort carbon scenario. From Fig. 15, we can see that the all requests go to the Ireland data center for all the time intervals. Fig. 16 depicts the number of requests serviced at each data center during a time interval for the best-effort electricity scenario. From Fig. 16, we can see that the California data center services all the requests for the first part of the time period and in second part the electricity costs of the Ireland and California data center become almost equal so that some of the requests go to the California data center and some of the requests go to the Ireland data center. Finally, all the requests get serviced by the California data center as the prices diverge.

Fig. 17 depicts the carbon emitted under each scenario over the time period. From Fig. 17 we can see that the carbon emitted follows the number of service requests

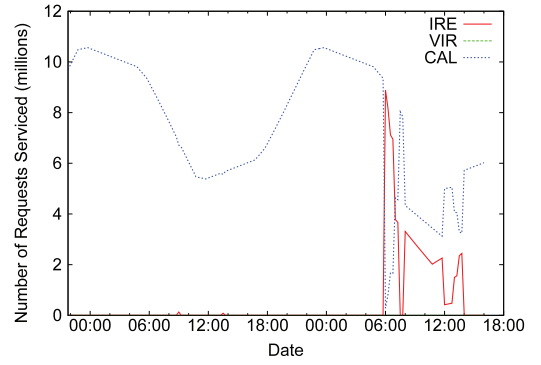


Fig. 16. Number of requests serviced at each data center when the “Best Effort Electricity” scenario is used. The time zone of the  $x$ -axis is GMT.

under all scenarios. This is as we expected as more service requests require more energy which increases the carbon emissions. We can also see there are no spikes in the emissions which is as we expected as Fig. 6 shows that the change in carbon intensity over time is gradual. Finally, we can see that the difference between the schemes is quite small. The carbon intensities of the three data centers are relatively similar. Ireland’s data center carbon intensity ranges from 369 g/kWhr to 522 g/kWhr, Virginia has a carbon intensity of 559 g/kWhr and California has a carbon intensity of 384 g/kWhr, but this is offset by the cooling setup used in our simulation data center. The difference in carbon intensities between other regions is much larger. For example Norway with its high level of hydropower has a carbon intensity of 6 g/kWhr, while Austria has a carbon intensity of 870 g/kWhr.

Fig. 18 depicts the electricity cost under each scenario over the time period. From Fig. 18 we can see that the difference between the various scenarios can be quite large particularly when there are peaks in the electricity cost at the data center where requests are being serviced. It is also interesting to note that the difference can be quite small as we can see that the electricity cost of the best-effort carbon scenario and the best-effort electricity scenario are effectively the same toward the end of the time period. Between 05:00 and 07:00 of the second day in Fig. 18 we can see that the electricity cost of the best-effort carbon and best-effort electricity schemes are equal as the data center with the cheapest electricity also has the lowest carbon intensity during this time period. It should be noted, that while the

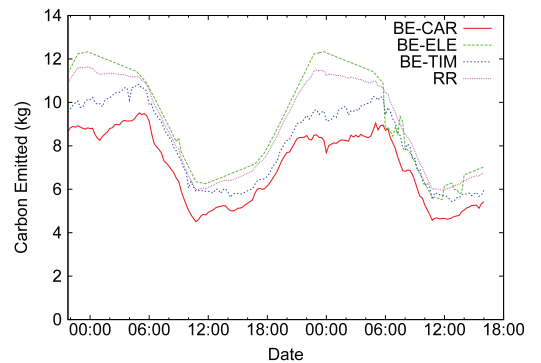


Fig. 17. Carbon Emitted at each time interval under a variety of scenarios. The time zone of the  $x$ -axis is GMT.

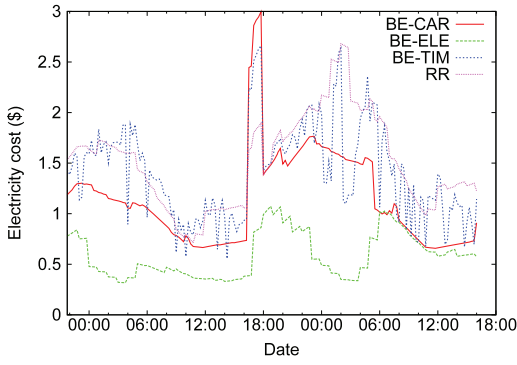


Fig. 18. Electricity Cost at each time interval under a variety of scenarios. The time zone of the  $x$ -axis is GMT.

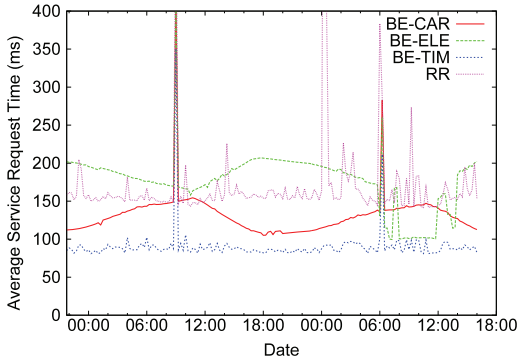


Fig. 19. Average service request time at each time interval under a variety of scenarios. The time zone of the  $x$ -axis is GMT.

overall electricity cost is quite low it is the relative differences in the electricity price that are the most important. In our model, we assumed that a single server performs all the computation required for a single request. While it is possible for cloud services to use this approach, latency consideration frequently result in a partition/aggregate design pattern being used [42]. In this approach, a request is broken into pieces which are then farmed out to worker servers. The responses of the workers are aggregated together by aggregator servers to yield the result to the request. In this design, hundreds of servers can be used to process a single request although typically tens of servers are used to handle requests. In this design the energy consumption for a request is significantly higher as tens of servers are operating simultaneously and the overall electricity cost would consequently be significantly higher. We chose not to model the requests in this fashion as without trace data it is difficult to simulate this design accurately as each worker server frequently operates for different lengths of time and therefore the energy consumed by each worker will be different.

Fig. 19 depicts the average service request time under the various scenarios. From Fig. 19 we can see that the best-effort carbon and best-effort electricity scenarios have average service request times that fluctuate smoothly. This is a result of not using the Virginia data center to service the load as its congestion causes the average service request time to change frequently. The reason for the change in average service request time for the best-effort carbon and best-effort electricity scenarios is the changes in demand in the regions. If there are more requests from a region with

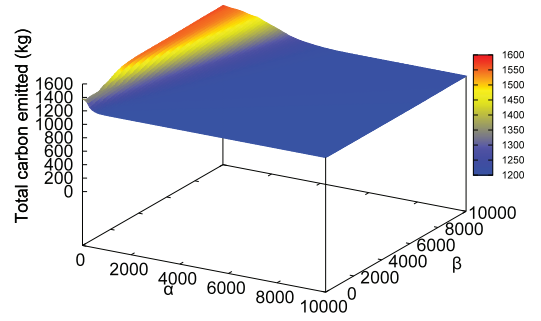


Fig. 20. Total carbon output with varying relative price functions.

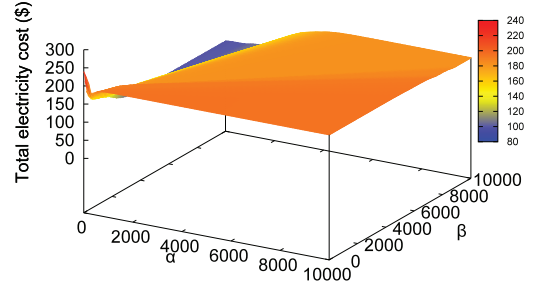


Fig. 21. Total electricity cost with varying relative price functions.

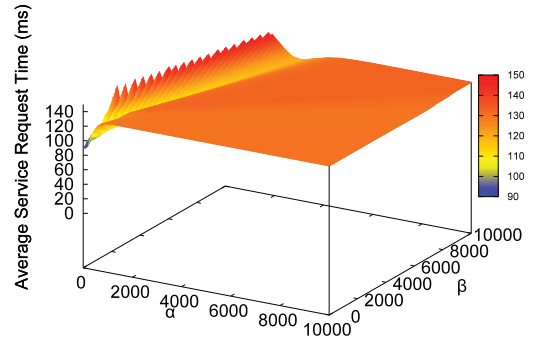


Fig. 22. Average service request time with varying relative price functions.

high latency at a given time interval, this will increase the average service request time. This does not occur in the roundrobin and best-effort time scenarios as the requests are spread among the three data centers and the effect is diluted. It is important that the cloud operator considers this so that SLAs are not violated.

We now examine the second set of simulations. Fig. 20 depicts the total carbon output as we vary  $\alpha$  and  $\beta$ . From Fig. 20 we can see that as we increase  $\alpha$  the carbon emissions decrease and that as we increase  $\beta$  the total carbon emissions increase. Fig. 21 depicts the total electricity cost as we vary  $\alpha$  and  $\beta$ . From Fig. 21 we can see that as we increase  $\alpha$  the electricity cost increases and as we increase  $\beta$  the electricity cost decreases. Fig. 22 depicts the average service request time as we vary  $\alpha$  and  $\beta$ . From Fig. 22 we can see that as we increase  $\alpha$  and  $\beta$  we increase the average service request time. The increase is more severe in  $\beta$ 's case, but this is a result of the particulars of the simulations as the average latency between all the regions and the California data center is higher than the average latency between all the regions and the Ireland data center. Fig. 21 is not monotonic as the initial increase of  $\alpha$  from zero will move requests from Virginia to Dublin which lowers the electricity cost, while

increasing  $\alpha$  beyond this point moves requests from California to Dublin which increases the overall electricity cost. Similar trends can be seen in Fig. 22.

The selection of  $\alpha$  and  $\beta$  are of crucial importance to the operation of this algorithm. Ultimately the selection of these values will depend on the SLAs the cloud operator has agreed to. Savings in electricity cost and reductions in carbon emissions can only be achieved if SLAs can still be maintained while there is an increase in the average service request time. The selection of whether to lower carbon emissions or electricity cost will depend on whether the cloud operator is under any regulation to limit its carbon emissions, public relations pressures or the price of carbon on carbon trading markets. The first set of simulations has shown the average service request time will vary with time. From this, we can conclude that  $\alpha$  and  $\beta$  will also vary with time. It would, however, be relatively simple to alter the algorithm so that  $\alpha$  and  $\beta$  are adjusted at each time interval in relation to the average service request time of the previous time interval and this is left for future work.

## 8 CONCLUSION

We have shown that a cloud can be operated in such a manner to lower carbon emissions and operational cost. Our simulations show that there is a corresponding penalty in terms of average service request time if the cloud is run in such a fashion. Our work examines the electricity cost, carbon emissions, and average service request time for a variety of scenarios. The decision concerning how to balance the various factors will depend on SLAs, government legislation, and the price of carbon on trading schemes. Using this information and the specifics of the cloud the operator can run the cloud in the most desirable fashion. The nature of the service will determine if a cloud owner can implement this algorithm while conforming to service level agreements.

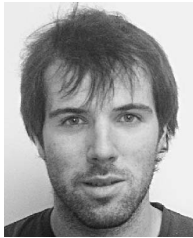
## ACKNOWLEDGMENTS

This work is partially funded by the Irish Higher Education Authority under the HEA PRTL Telecommunications Graduate Initiative and by Science Foundation Ireland grant 11/PI/1177.

## REFERENCES

- [1] A. Qureshi, J. Gutttag, R. Weber, B. Maggs, and H. Balakrishnan, "Cutting the Electric Bill for Internet-Scale Systems," *Proc. ACM SIGCOMM*, pp. 123-134, Aug. 2009.
- [2] R. Stanojević and R. Shorten, "Distributed Dynamic Speed Scaling," *Proc. IEEE INFOCOM*, pp. 1-5, Mar. 2010.
- [3] Z. Liu, M. Lin, A. Wierman, S.H. Low, and L.L. Andrew, "Greening Geographical Load Balancing," *Proc. ACM SIGMETRICS Joint Int'l Conf. Measurement and Modeling of Computer Systems (SIGMETRICS)*, pp. 233-244, June 2011.
- [4] M. Conti, E. Gregori, and F. Panzieri, "Load Distribution Among Replicated Web Servers: A QoS-Based Approach," *SIGMETRICS Performance Evaluation Rev.*, vol. 27, no. 4, pp. 12-19, 2000.
- [5] Z.M. Mao, C.D. Cranor, F. Bouglis, M. Rabinovich, O. Spatscheck, and J. Wang, "A Precise and Efficient Evaluation of the Proximity between Web Clients and their Local DNS Servers," *Proc. General Track of the Ann. Conf. USENIX Ann. Technical Conf. (USENIX)*, pp. 229-242, June 2002.
- [6] M. Pathan, C. Vecchiola, and R. Buyya, "Load and Proximity Aware Request-Redirection for Dynamic Load Distribution in Peering CDNs," *Proc. OTM Confederated Int'l Conf. on the Move to Meaningful Internet Systems*, pp. 62-81, 2008.
- [7] "Make IT Green Cloud Computing and Its Contribution to Climate Change," *Greenpeace*, Retrieved, <http://www.greenpeace.org/international/Global/international/planet-2/report/2010/3/make-it-green-cloud-computing.pdf>, Feb. 2011.
- [8] I.B. Fridleifsson, R. Bertani, E. Huenges, J.W. Lund, A. Ragnarsson, and L. Rybach, "The Possible Role and Contribution of Geothermal Energy to the Mitigation of Climate Change," *Proc. IPCC Scoping Meeting on Renewable Energy Sources Conf.*, pp. 59-80, 2008.
- [9] M. Lenzen, "Life Cycle Energy and Greenhouse Gas Emissions of Nuclear Energy: A Review," *Energy Conversion and Management*, vol. 49, no. 8, pp. 2178-2199, Aug. 2008.
- [10] "European Union Emissions Trading System," <http://ec.europa.eu/clima/policies/ets/>, 2013.
- [11] C. Ren, D. Wang, B. Urgaokar, and A. Sivasubramaniam, "Carbon-Aware Energy Capacity Planning for Data Centers," *Proc. IEEE MASCOTS*, pp. 391-400, Aug. 2012.
- [12] Z. Liu, M. Lin, A. Wierman, S.H. Low, and L.L. Andrew, "Geographical Load Balancing with Renewables," *Proc. GreenMETRICS*, pp. 1-5, June 2011.
- [13] J. Doyle, D. O'Mahony, and R. Shorten, "Server Selection for Carbon Emission Control," *Proc. ACM GreenNet*, pp. 1-6, Aug. 2011.
- [14] J. Baliga, R.W.A. Ayre, K. Hinton, and R.S. Tucker, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," *Proc. IEEE*, vol. 99, no. 1, pp. 149-167, Jan. 2011.
- [15] J.W. Durham, R. Carli, P. Frasca, and F. Bullo, "Discrete Partitioning and Coverage Control for Gossiping Robots," *IEEE Trans. Robots*, vol. 28, no. 2, pp. 364-378, 2012.
- [16] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing Electricity Cost: Optimization of Distributed Internet Data Centers in a Multi-Electricity-Market Environment," *Proc. IEEE INFOCOM*, pp. 1-9, Mar. 2010.
- [17] L. Rao, X. Liu, M.D. Ilic, and J. Liu, "Distributed Coordination of Internet Data Centers under Multiregional Electricity Markets," *Proc. IEEE*, vol. 100, no. 1, pp. 269-282, Jan. 2012.
- [18] P. Wang, L. Rao, X. Liu, and Y. Qi, "D-Pro: Dynamic Data Center Operations with Demand-Responsive Electricity Prices in Smart Grid," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 1743-1754, Dec. 2012.
- [19] V. Mathew, R.K. Sitaraman, and P. Shenoy, "Energy-Aware Load Balancing in Content Delivery Networks," *Proc. IEEE INFOCOM*, pp. 954-962, Mar. 2012.
- [20] P. Mahadevan, S. Banerjee, and P. Sharma, "Energy Proportionality of an Enterprise Network," *Proc. ACM GreenNet*, pp. 53-60, Aug. 2010.
- [21] F.F. Moghaddam, M. Cheriet, and K.K. Nguyen, "Low Carbon Virtual Private Clouds," *Proc. IEEE Int'l Conf. Cloud Computing*, pp. 259-266, July 2011.
- [22] P.X. Gao, A.R. Curtis, B. Wong, and S. Keshav, "It's Not Easy Being Green," *Proc. ACM SIGCOMM*, pp. 221-222, Aug. 2012.
- [23] F. Aurenhammer, "Voronoi Diagrams-A Survey of a Fundamental Geometric Data Structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345-405, Sept. 1991.
- [24] R.J. Fowler, M.S. Paterson, and S.L. Tanimoto, "Optimal Packing and Covering in the Plane Are NP-Complete," *Information Processing Letters*, vol. 12, pp. 133-137, 1981.
- [25] R.Z. Hwang, R.C.T. Lee, and R.C. Chang, "The Slab Dividing Approach to Solve the Euclidean P-Center Problem," *Algorithmica*, vol. 9, pp. 1-22, 1993.
- [26] T.F. Gonzalez, "Clustering to Minimize the Maximum Intercluster Distance," *Theoretical Computer Science*, vol. 38, pp. 293-306, 1985.
- [27] "Elastic Compute Cloud," *Amazon*, <http://aws.amazon.com/ec2>, 2013.
- [28] "Carbon Monitoring for Action," <http://carma.org/>, 2013.
- [29] "eGRID," *United States Environmental Protection Agency*, <http://www.epa.gov/cleanenergy/energy-resources/egrid/index.html>, 2013.
- [30] *Eirgrid*, <http://www.eirgrid.com>, 2013.
- [31] M. Pervilä and J. Kangasharju, "Cold Air Containment," *Proc. ACM GreenNet*, pp. 7-12, Aug. 2011.
- [32] L.A. Barroso and U. Hölzle, "The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines," *Synthesis Lectures on Computer Architecture*, Morgan and Claypool Publishers, 2009.

- [33] D. Atwood and J.G. Miner, "Reducing Data Center Cost with an Air Economizer," <http://www.intel.com/content/www/us/en/data-center-efficiency/data-center-efficiency-xeon-reducing-data-center-cost-with-air-economizer-brief.html>, Aug. 2008.
- [34] P. Rumsey, "Overview of Liquid Cooling Systems," [http://hightech.lbl.gov/presentations/Dominguez/5\\_LiquidCooling\\_101807.ppt](http://hightech.lbl.gov/presentations/Dominguez/5_LiquidCooling_101807.ppt), 2007.
- [35] A. Almoli, A. Thompson, N. Kapur, J. Summers, H. Thompson, and G. Hannah, "Computational Fluid Dynamic Investigation of Liquid Rack Cooling in Data Centres," *Applied Energy*, vol. 89, pp. 150-155, 2012.
- [36] M.G. Corporation, "Flovent Version 9.1," Wilsonville, Oregon, USA, 2010.
- [37] R.K. Sharma, C.E. Bash, and C.D. Patel, "Balance of Power: Dynamic Thermal Management for Internet Data Centers," *IEEE Internet Computing*, vol. 9, no. 1, pp. 42-49, Jan./Feb. 2005.
- [38] J. Moore, J.S. Chase, P. Ranganathan, and R. Sharma, "Making Scheduling 'Cool': Temperature-Aware Workload Placement in Data Centers," *Proc. Ann. Conf. USENIX Ann. Technical Conf. (USENIX)*, pp. 61-75, Apr. 2005.
- [39] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: An Overlay Testbed for Broad-Coverage Services," *ACM SIGCOMM Computer Comm. Rev.*, vol. 33, no. 3, pp. 3-12, 2003.
- [40] "Social Bakers the Recipe for Social Marketing Success," <http://www.socialbakers.com/facebook-statistics/>, 2013.
- [41] *Internet World Stats*, "Internet World Stats Usage and Population Statistics," <http://www.internetworldstats.com>, 2013.
- [42] M. Alizadeh, A. Greenberg, D.A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data Center TCP (DCTCP)," *Proc. ACM SIGCOMM*, pp. 63-74, Aug./Sept. 2010.



**Joseph Doyle** received the BA degree in computer and electronic engineering as a gold medallist, the BAI degree in 2009, and the PhD degree all from Trinity College Dublin, in 2013. His research interests include cloud computing, green computing, cloud gaming, and network optimization. He is currently a postdoctoral researcher in Trinity College Dublin.



**Robert Shorten** received the first class honours BE degree in electronic engineering from the University College Dublin (UCD), in 1990, and the MEngSc and PhD degrees from the UCD, in 1993 and 1996, respectively. From 1993 to 1996, he was the holder of a Marie Curie Fellowship to conduct research at the Daimler-Benz Research Institute for Information Technology in Berlin. In 1996, he was invited as a visiting fellow to the Center for Systems

Science, Yale University, to work with Professor K.S. Narendra. Since returning to Ireland in 1997 as the recipient of a European Presidential Fellowship he has built up a research group in the area of switching systems. He is an editor of the IEE Proceedings on Control, and is a cofounder of the Hamilton Institute, National University of Ireland, Maynooth. He is currently a senior research manager at IBM Research in Ireland.



**Donal O'Mahony** is a professor of computer science at Trinity College Dublin. His research interests include the challenges of delivering a future Internet that can provide services to a highly connected society with reliability and security. Over the years, this had led to explorations into cryptography, network protocols, security applications, mobility, ad hoc networking and the software engineering of networked systems. In 2004, he founded the

CTVR telecommunications research centre which in turn acted as the catalyst for the establishment of Bell Labs Ireland. Under his stewardship, this center raised some 25 million of research funding and published more than 400 high-quality papers in international journals and conferences. He now leads the Networks and Telecommunications Research Group at Trinity College where he has authored two books and more than 100 papers. He is a fellow of Trinity College Dublin (2000) and a Fulbright fellow (1999).

► **For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).**