A Fuzzy Logic-based Controller for Cost and Energy Efficient Load Balancing in Geo-Distributed Data Centers

Adel Nadjaran Toosi and Rajkumar Buyya
Cloud Computing and **D**istributed **S**ystems (CLOUDS) Laboratory
Department of Computing and Information Systems
The University of Melbourne, Australia
Email: {anadjaran,rbuyya}@unimelb.edu.au

***Abstract*—The ever-increasing demand for cloud services results in large electricity costs to cloud providers and causes significant impact on the environment. This has pushed cloud providers to power their data centers with renewable energy sources more than ever. Among the different ways of adopting renewable energy sources, on-site power generation using wind and solar energy has gained considerable attention by large companies and proved its potential to reduce data centers' carbon footprint and energy costs. Efficient utilization of renewable energy sources is challenging due to their intermittency and unpredictability. Cloud providers with multiple Geo-distributed data centers in a region can exploit the temporal variations in on-site power and grid power price by routing the load to a suitable data center in order to reduce cost and increase renewable energy utilization. To achieve this goal, we propose a fuzzy logic-based load balancing algorithm that acts with no knowledge of future. We conduct extensive experiments using a case study based on real world traces obtained from National Renewable Energy Laboratory (NREL) and Energy Information Administration (EIA) in the US, and Google cluster-usage. Compared to other benchmark algorithms, our method is able to significantly reduce the cost without a priori knowledge of the future electricity price, renewable energy availability, and workloads.**

I. Introduction

Data centers used for hosting cloud applications are known to be consuming large amount of electricity leading to high operational cost for the cloud providers and high carbon footprint on the environment. According to a report from NRDC[1] [1], in 2013, US data centers alone consumed 91 billion kilowatt-hours of electricity, equivalent to two-year power consumption of all households in New York city. This is projected to increase to roughly 140 billion kilowatt-hours and is responsible for the emission of nearly 150 million metric tons of carbon pollution per annum in 2020. Accordingly, cloud service providers are working hard to reduce their energy consumption and their dependence on power generated from fossil fuels (i.e., Brown energy) having adverse impact on the environment.

Companies such as Google[2], Microsoft[3] and Amazon[4] are working towards this goal by using renewable energy sources (i.e., Green energy) to power their data centers and making direct investments in on-site green power generation. Photovoltaic solar panels that directly convert sunlight into electricity and wind turbines that capture wind energy and turn it into electricity are among the most popular on-site power sources used by contemporary data centers. For example Amazon Web Services (AWS) is building a wind farm that will be operational by end-2016 that generates 40 percent of its electrical usage from renewable energy sources[5]. Powering data centers entirely with these renewable energy sources is a challenging issue due to the intermittency and unpredictability of wind and solar energy. Therefore, to serve their users, providers end up using grid power or brown energy, besides their on-site renewable energy sources, in data centers. Yet, to minimize brown energy usage, they need to obtain the highest possible renewable energy utilization.

There are considerable number of studies illustrating the potential of using "*geographical load balancing*" (GLB) in reducing brown energy usage and accordingly maximizing renewable energy utilization [2], [3], [4]. GLB provides such an opportunity for providers having multiple geographically distributed data centers by allowing for "*follow the renewables*" routing [3]. Additionally, GLB also allows for routing the load to places with lower grid power prices even if renewable energy sources are currently fully utilized. This eventually leads to significant cost savings for the cloud provider.

Although the GLB approach benefits cloud providers significantly, it raises an interesting, and challenging question: "with limited or even no a priori knowledge of the future workload, and dynamic and unpredictable nature of renewable energy sources, how does one allocate requests to each data center such that the total cost of power consumption is minimized and accordingly the overall renewable energy usage is maximized? (see Figure 1).

In this paper, we aim to address the above practical and yet fundamental question. First, we characterize the optimal offline load balancing algorithm in which, to a certain time window, future knowledge of renewable energy traces and workload (i.e., arriving time and duration of requests) of each data center site are assumed to be known. Then, we show that the optimal strategy is computationally prohibitive and suffers

[1]Natural Resources Defense Council, www.nrdc.org.
[2]http://www.google.com.au/green/energy/
[3]http://www.microsoft.com/environment/renewable.aspx/.
[4]http://aws.amazon.com/about-aws/sustainable-energy/.
[5]http://www.reuters.com/article/2015/07/14/us-amazon-iberdrola-idUSKCN0PO1PF20150714

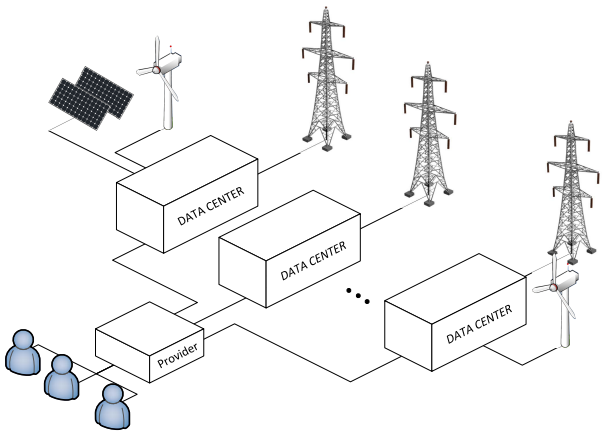


Fig. 1. System Model.

from the "*curse of dimensionality*" [5]. This makes the optimal load balancing computationally intractable, particularly when the future window size, the number of data center sites, and number of requests grow. It is in fact considerably difficult to balance the load optimally, even when the exact knowledge about the future is available.

To deal with the complexity, we propose a *fuzzy logic-based* heuristic for GLB that optimizes the overall cost and the renewable energy utilization with no future knowledge. The fuzzy logic-based load balancing method works based on the recent history of the load and availability of renewable energy sources in data centers. It routes requests according to comparison between the suitability values, each calculated as an output of a fuzzy inference engine for the corresponding data center.

The fuzzy inference engine, as input, uses the utilization of renewable energy and the amount of brown energy usage for the data center, along with the average electricity price in the data center's location within a certain time window of a recent history. According to fuzzy rules, the fuzzy engine calculates a value illustrating the suitability of the data center for accommodating the current request. The fuzzy logic-based load balancing method uses all suitability values calculated in this way to make the final decision regarding routing a request.

The **main contributions** of the paper are:

1) We characterize the optimal offline GLB, in which the exact future demand is assumed to be known a priori and show that the optimal strategy is computationally intractable.
2) We propose a fuzzy logic-based load balancing algorithm that provides a significant cost savings without any knowledge of future demands, renewable energy sources, and prices in electricity market beforehand.
3) We evaluate our proposed algorithm using simulation of a case study designed according to the real world traces of meteorological data and electricity prices under real data center workloads.

The rest of the paper is organized as follows: Section II describes the system model and formally defines the GLB problem. We introduce optimal load balancing algorithm in Section III and we discuss its intractability. In Section IV, we propose fuzzy logic-based load balancing after a brief introduction to fuzzy systems. The performance evaluation of the proposed strategy and comparison with benchmark algorithms are presented in Section V. Finally, our conclusions and future work are presented in Section VII.

II. System Model

We focus on how to dynamically provision the service capacity in geographically dispersed data centers serving user requests arriving in a single entry point (e.g., a provider portal's interface) such that both the overall "utilization of the on-site renewable energy sources" is maximized and the monetary "cost" of the system is minimized. First, we introduce a general model for this setting, and then we formulate the GLB problem.

A. Preliminaries

Most cloud providers offer their services in multiple locations world-wide called *regions*. For example, Amazon Web Services (AWS)[6] currently has data centers located in 9 different regions; three in *Asia Pacific*, two in *Europe*, one in the *US east*, two in the *US west*, and one in *South America*. Each region has a collection of n isolated data centers D within the region, often called *zone*. These data centers are located in geographically diverse locations within a region and each uses various sources of energy. In our model, we assume data center $d_i \in D$ $(1 \leq i \leq n)$ has two main sources of electricity: power generated by local on-site renewable energy sources (e.g., solar, wind or mixture) and utility grid.

Suppose that the cloud provider receives a set of m service requests $R = \{r_1, ..., r_m\}$, in the specific time window, from cloud users who do not have preferences over data centers in the region. A request can vary from the one submitted for acquiring container-based instances or hypervisor-based virtual machines (VMs) to a job-based request for running an application in the specific time period. A request r_j $(1 \leq j \leq m)$ is denoted by three-tuple (a_j, q_j, l_j), where a_j is the arrival time of the request within the time window, q_j is the required quantity of processing units (e.g., total number of ECUs[7] for VMs), and l_j is the lifetime of the request (i.e., the holding time of VMs). In practice, the lifetime and arrival time of a request is not known by the provider in advance, e.g., a cloud provider does not know when next VM request arrives and how long the user will hold allocated VMs. Note that, in our model, requests must be served based on the order of arrival in a *first come first served* fashion. No request can be delayed in favor of another request.

We consider a discrete-time model in which the amount of power generated by on-site renewable energy sources per each time slot (e.g., kWh) is given for every data center. Let $e_{i,t}$ denote the units of power generated from renewable sources at time slot t in data center i and not consumed by currently accommodated requests. The value of $e_{i,t}$ dynamically changes based on the weather condition and availability of renewable

[6]Amazon Web Services, http://aws.amazon.com/.

[7]In Amazon Web Services (AWS) terminology, 1 EC2 Compute Unit (ECU) is equivalent to a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor.

energy sources on the location and is assumed to be difficult to predict. The cost of power generated from renewable sources is considered to be zero for the cloud provider. However, if power supply from renewable sources is not enough to cover the demand in the data center, the shortage will be supplied from the grid. We make an assumption that the data center buys electricity from a wholesale market and the spot price of electricity varies over time according to the location of the data center. Let $p_{i,t}$ be the price of electricity at the time slot t at the location of data center i.

We assume that by accommodating a request, the cloud provider on average consumes a constant rate of energy per time slot to fulfill the user requirement (e.g., 100 Watt-hour). Therefore, the provider with x units of available renewable energy at time slot t is only able to accommodate up to x requests with unit quantity demand and unit power usage per time slot for free.

For the sake of simplicity, we assume that values of q_j, $e_{i,t}$ and $p_{i,t}$ are reported according to the rate of energy consumption per smallest unit of demand, e.g., q_j shows the total units of power consumption by a request.

B. Geographical Load Balancing Problem

Cloud users often do not have preference over different zones (data centers) in a region. This gives the cloud provider an opportunity to route users' requests to data centers with higher availability of renewable energy sources and lower market price of electricity. The problem is to optimally distribute requests among data centers such that the overall renewable energy consumption is maximized and the total incurred cost is minimized.

Suppose that the cloud provider offers its services in multiple data centers $D = \{d_1, d_2, ..., d_n\}$ in a given region. Let $s_{i,t}$ denote the total number of computing units consumed by active requests (i.e., not finished) accommodated in data center i at time t, that is:

$$s_{i,t} = \sum_{r_j \in R_{i,t}} q_j \tag{1}$$

where $R_{i,t}$ is the set of requests that are active in data center i at time t.

Suppose that the cloud provider can precisely predict the following values for every time slot t in a future time window of size T ($1 \leq t \leq T$):

1) $R = \{r_1, ..., r_m\}$ list of requests arriving in different time slot,
2) the arrival time a_j, quantity q_j, and lifetime l_j of the request r_j,
3) $e_{i,t}$, the total units of power generated from renewable sources at time t in each data center, and
4) $p_{i,t}$, the spot price of electricity per unit power at time t in the location of each data center.

Assuming that the future knowledge is available for all the aforementioned values, the load balancing problem can be defined as the minimization problem of the total cost C computed as:

$$C = \min \sum_{i=1}^{n} \sum_{t=1}^{T} p_{i,t} \times (s_{i,t} - e_{i,t})^{+}, \tag{2}$$

where $(x)^{+} = \max(0, x)$. Equation (2) minimizes the total cost of power usage assuming that perfect future knowledge in a window of size T is available.

The main challenge of problem (2) is that how to optimally redirect arriving requests to different data centers $d_i \in D$ such that the total cost of the grid power usages is minimized.

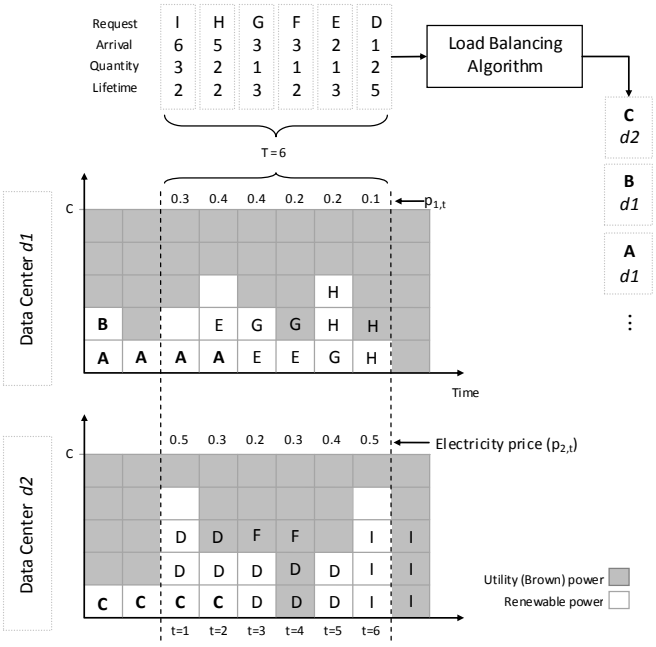


Fig. 2. An example of optimal Geographical Load Balancing.

Figure 2 illustrates an example of optimal GLB with two data centers and a future time window of size 6. A set of 6 requests labeled D to I with shown configurations arrive within the window. The availability of renewable power units (non-shaded cells) and electricity price per each time slot for both data centers are also shown. The figure depicts the optimal way of distributing requests between the data centers, which leads to the minimal cost of 1.7 within the window. The total cost is computed according the summation of the price for power units consumed in the shaded units. The optimal solution is obtained by checking all possible combinations of assigning the 6 requests to different data centers. Interested readers might check that no other allocations lead to a total cost lower than 1.7. Note that earlier received requests A to C are optimally assigned to data centers in former rounds of the optimization process; so they are not considered in the cost calculation and only affect the availability of renewable energy.

III. Optimal Geographical Load Balancing and Its Intractability

In this section, we derive the optimal solution for the GLB problem (explained in the previous section) and show that such

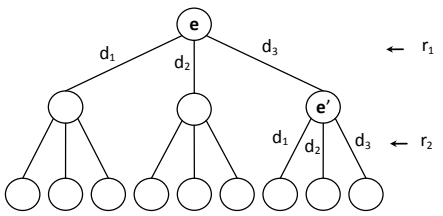


Fig. 3. Decision tree for Geographical Load Balancing.

an optimal solution is computationally intractable. The optimal solution can be obtained by constructing a *decision tree* that covers all possible combinations.

We start with an initial state including renewable energy units available in all data center. For every arriving request within the window, there are n different possible placement options (i.e., the number of data centers in the region), which specifies the branching factor in our decision tree. Nodes of the tree keep the state and branches change the state by accommodating the associated requests in different data centers. We expand the tree by placing upcoming requests in all possible data centers presuming earlier requests are located according to the placement in upper levels of the tree. The process continues until all arriving requests in the window are covered. A path from the root to a leaf of the tree that generates the lowest cost determines the optimal solution. Figure 3 illustrates the process.

Let $R = \{r_1, ..., r_m\}$ be the list of requests arriving in future time window ($a_m \leq T$) sorted according to the arrival time of requests, i.e., $a_1 \leq a_2 \leq a_3 \leq ... \leq a_m$. Placing request r_j in data center d_i will change the availability of renewable power units $e_{i,t}$ during the lifetime of the request. We define a state variable $\mathbf{e}$ as an n-tuple of T-sets $\mathbf{e} = \{\{e_{1,1}, .., e_{1,T}\}, ..., \{e_{n,1}, .., e_{n,T}\}\}$, where every set $e_i = \{e_{i,1}, .., e_{i,T}\}$ shows the availability of renewable power units for data center i ($1 \leq i \leq n$) for the period of $[1..T]$. The cost of accommodating a request r_j in data center d_i, when the state is $\mathbf{e}$ can be computed as follows:

$$c(r_j, d_i) = \sum_{t=a_j}^{\max(T, a_j + l_j - 1)} (q_j - e_{i,t})^+ \times p_t \,, \qquad (3)$$

where $\max(T, a_j + l_j - 1)$ assures that cost calculation is done within the window and $(q_j - e_{i,t})^+$ counts the total units of brown energy used by the request r_j per each time slot.

Algorithm 1 shows the pseudo-code of the recursive function for the minimum cost calculation in the window based on the availability of renewable energy and electricity prices for each time slot. Lines 8 to 11 compute the cost incurred by placing request r_j in data center d_i. The outer loop in Line 6 checks every possible placement option for request r_j and finds the minimum cost by a recursive call for the remaining requests at Line 12.

Algorithm 1 is *brute-force* search and makes a decision tree by branching factor n upto m levels. This leads to the total number of $1 + n + n^2 + n^3 + ...n^{m+1} = \frac{n^{m+1}-1}{n-1}$ function calls, i.e, $O(n^m)$ different state variables. Hence, the overall computational complexity of algorithm is $O(n^m)$ that offers an *exponential time* solution for Equation (2), given

Algorithm 1 Optimal Geographical Load Balancing Algorithm

Input: $j, \mathbf{e}$
Output: $mincost$
1: **function** GLB-OPT($j, \mathbf{e}$)
2: **if** $j > T$ **then**
3: **return** 0
4: **end if**
5: $mincost \leftarrow +\infty$
6: **for** $i \leftarrow 1$ **to** n **do**
7: $\mathbf{e}' \leftarrow \mathbf{e}$
8: **for** $t \leftarrow a_j$ **to** $\min(T, a_j + l_j - 1)$ **do**
9: $c(r_j, d_i) \leftarrow (q_j - e'_{i,t})^+ \times p_{i,t}$
10: $e'_{i,t} \leftarrow (e'_{i,t} - q_j)^+$
11: **end for**
12: $cost \leftarrow c(r_j, d_i) +$ GLB-OPT$(j + 1, \mathbf{e}')$
13: **if** $cost < mincost$ **then**
14: $mincost \leftarrow cost$
15: **end if**
16: **end for**
17: **return** $mincost$
18: **end function**

a bounded number of data centers. Given that, in practice, m could be considerably large even for small size window (T), the algorithm is computationally intractable. This time complexity can be improved using techniques such as branch and bound. However, it does not reduce the worst-case time complexity. If the lifetime of requests is bounded (i.e., $a_j < k$) and is considerably smaller than the window (i.e., $k \ll T$), the time complexity can also be improved proportional to the difference between k and T by using a dynamic programming technique. We do not discuss details of such algorithms further here.

IV. Fuzzy logic-based load balancing

In the previous section, we showed that the optimal GLB is hard to achieve, even the prefect knowledge of renewable energy traces, electricity prices, and future demands are known in advance. In this section, we present a fuzzy logic-based load balancing algorithm that only uses recent data history to optimize the load balancing problem. First, we provide a brief introduction to fuzzy logic systems and why we selected fuzzy control. Then, we explain our proposed *fuzzy logic-based load balancing* (FLB) method to address the problem of GLB with no future knowledge.

A. Fuzzy logic systems

In general, a fuzzy logic system is a reasoning structure that provides a means for converting linguistic strategies into control decisions. By using simple linguistic rules, it can attain a nonlinear mapping of an input space to an output space. For example, with information about how fast you are driving and how close the object is, a fuzzy logic system can accordingly provide required pressure on the brakes.

Figure 4 depicts a typical fuzzy inference system. The inputs are transformed into fuzzy sets using the *fuzzification* module. For example, the height of 180cm, as an input, is converted to a fuzzy set specifying the degree of *tallness*. A fuzzy set is characterized by a *Membership Function* (MF)

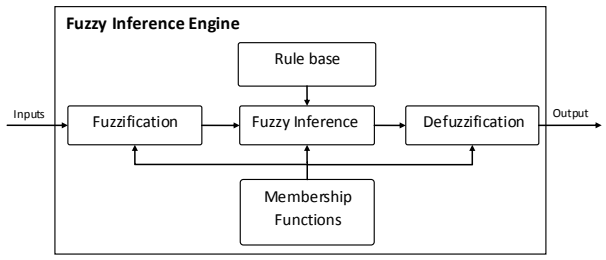


Fig. 4. A typical fuzzy inference system.

specifying to what degree a value belongs to the corresponding fuzzy set. In a fuzzy inference system, each input and output variable has its own set of membership functions. The *fuzzy Inference* module using a set of fuzzy rules defined in the *rule-base* maps an input space to an output space. Fuzzy rules are defined in form of "IF-THEN" rules by a designer to relate input with output variables. For example a rule can be defined as:

if *height* is **very high** and *weight* is **very low** then *healthiness* is **low**.

The *defuzzification* module transforms the aggregated fuzzy set generated by inference module into a crisp value, e.g, 20% for the *healthiness*.

Fuzzy logic is conceptually easy to understand and mathematical concepts behind fuzzy reasoning, even though subtle, are very simple. Moreover, fuzzy logic systems are good at dealing with uncertainty and lack of perfect information. In non-linear systems with an arbitrary complexity and large number of inputs, fuzzy logic reasoning is among the best applicable techniques [6]. It is difficult to find a mathematical solution for our cost optimization problem where various and complex parameters are affecting decisions regarding request routing in GLB and there is no exact future knowledge available. Many research studies (e.g., [2], [3]) tackled the GLB problem using optimization techniques such as convex optimization, Integer programming, etc. Most of these techniques require accurate future knowledge of renewable energy availability, while achieving exact future knowledge is not easy if not possible. Moreover, such optimization techniques provide certain solutions without considering imposed uncertainty and errors of predicted values.

We choose fuzzy logic to tackle our problem as it is simple and is one of the most effective ways in handling the uncertainty and multiple input variables. In addition, fuzzy logic-based systems can be easily tuned using expert knowledge and manipulating rules or fuzzy sets. In the next section, we propose a fuzzy logic-based technique to tackle the complex and intractable problem of GLB. To the best of our knowledge, we are the first to propose fuzzy logic-based controller for the GLB problem.

B. Fuzzy logic-based load balancing

Our proposed fuzzy logic-based load balancing method works based on a simple and intuitive idea that arriving requests must be redirected to the most suitable data center such that the least cost incurs. In order to achieve this goal, we use a fuzzy inference engine designed to give a suitability value for data center i according to the following input values computed within a window of size W in the recent history:

TABLE I. FUZZY ASSOCIATE MEMORY FOR FUZZY INFERENCE ENGINE RULES.

U_i	B_i	F_i	*Suitability*
low	low	-	veryhigh
low	high	-	high
mid	low	-	high
mid	high	-	mid
high	low	low	mid
high	low	mid	mid
high	low	high	low
high	high	low	mid
high	high	mid	low
high	high	high	verylow

1) *The utilization of renewable energy sources (U_i)*: is a value in the range $[0, 1]$ and is computed as the ratio of the number of used to the total number of available renewable power units.
2) *Amount of brown energy consumption (B_i)*: is a value in the range $[0, +\infty)$ and is computed as the ratio of the total number of units of brown power units used to the total number of available renewable power units.
3) *Average price of electricity in the location (F_i)*: is the average price for an unit of power.

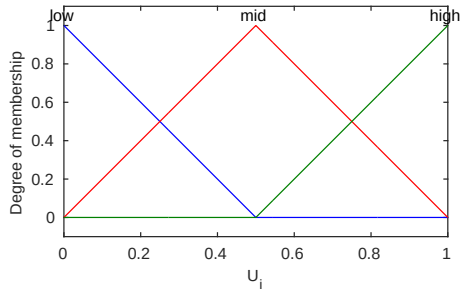
The fuzzy inference module based on the above inputs determines the suitability value, as an ouput, for each data center. We used a *Mamdani fuzzy inference* system [6] with *centroid of area* defuzzification strategy. All the MFs are triangular functions defined as shown in the Figure 5. The proposed fuzzy inference module uses the rules shown in the fuzzy associative memory in Table I. As an instance, the rule in the fifth row of the table can be interpreted as follows:

If U_i is **high** and B_i is **low** and F_i is **low** then *suitability* is **mid**.

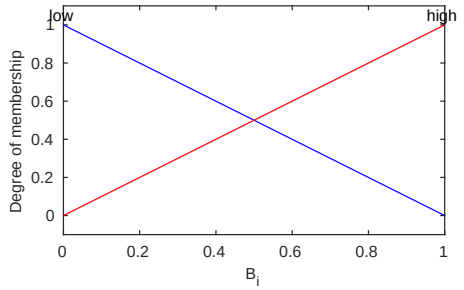
The output of the fuzzy inference engine, which varies between 0 and 1, specifies the suitability of each data center for routing the arriving request, where 0 shows the lowest suitability and 1 shows the highest suitability. Figure 5(d) shows MFs for the output of the fuzzy inference engine. The proposed fuzzy logic-based load balancing method computes the suitability value for all data centers and route the current request to a data center with the highest value of suitability. It finds the firing level of each rule (the degree to which the rule matches the inputs) and then calculates the output of each rule using fuzzy operators. Finally, the engine aggregates the individual rules outputs and obtains the overall fuzzy output of the system. The fuzzy output is converted to a crisp value by means of the defuzzification strategy.

V. PERFORMANCE EVALUATION

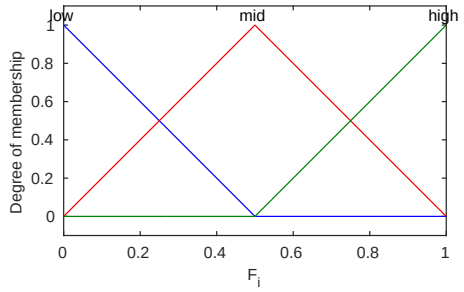
We conducted experiments based on simulation to study the performance of fuzzy logic-based load balancing (FLB). Our aim is to understand the renewable energy utilization and cost performance of FLB in realistic settings. In order to achieve our goal, we consider a case study based on real-world traces for the workload, renewable availability, and electricity prices.



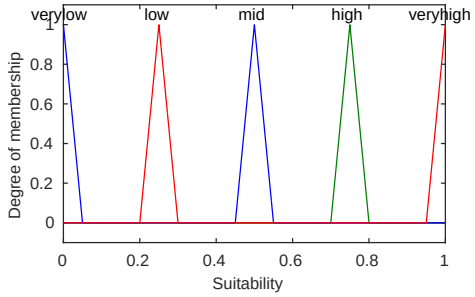
(a) Input MFs for the Utilization of Renewable Energy (U_i)



(b) Input MFs for the Brown Energy Consumption (B_i)



(c) Input MFs for the Price of Electricity (F_i)



(d) Output MFs for Suitability

Fig. 5. Fuzzy Engine Input and Output Fuzzy Sets.

A. Experimental Setup

1) Workload Setup: In order to generate IaaS workload for our case study, we use traces of Google cluster-usage [7]; as no publicly available workload trace of real-world IaaS clouds currently exists that we know of. This dataset includes the resource requirements of *tasks* submitted by 933 users to a Google cluster of 12K physical servers over a time period of 29 days. In our settings, we are interested in a workload

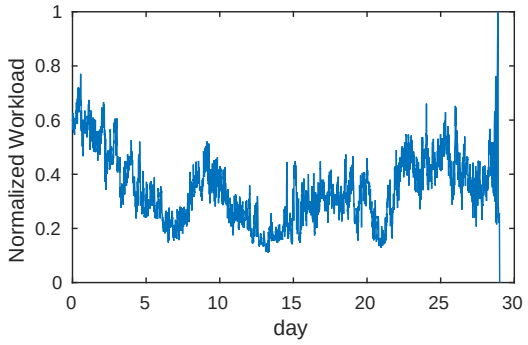


Fig. 6. Normalized workload.

containing requests for IaaS cloud services; however, Google dataset lacks such information. We therefore generate requests for each user in the Google dataset as if the user runs the tasks in an IaaS cloud environment such as EC2. In this regard, it is worth mentioning that in the Google cluster, tasks of different users might be scheduled onto a single server, while in a public IaaS cloud, tasks are executed in the user's allocated capacity environment (e.g., tasks are executed in a VM or a container).

Traces include records of submitted tasks each of which has resource requirements related to CPU, memory and disk [7]. As 93% of the Google cluster physical servers have the same computing capability, we align our capacity unit to that of a physical server in the cluster. For each user, we use a scheduling algorithm that assigns or releases capacity unit (e.g. a VM or a container) based on the resource requirements of the tasks. Whenever a user submits a task, the scheduling algorithm checks if it can accommodate the task in the currently allocated capacity; otherwise it allocates a new capacity unit. The scheduling algorithm also releases a capacity unit when there is no running task using that unit. As a result, we obtain list of requests for acquiring and releasing units of capacity in the data center and create a trace of roughly 470K requests. The normalized workload generated according to the explained method using the one-month Google cluster traces is illustrated in Figure 6. We assume that the power consumed by the unit capacity is on average 250W [8].

2) Configuration of data centers: We consider a group of 3 data center sites for the US-Southwest region in the following locations: *Prewitt* in New Mexico (NM), *Phoenix* in Arizona (AZ) and *Los Angeles* in California (CA). Data centers locations have been selected based on the availability

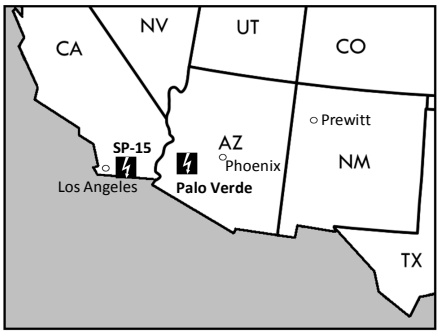


Fig. 7. Data center and hub locations.

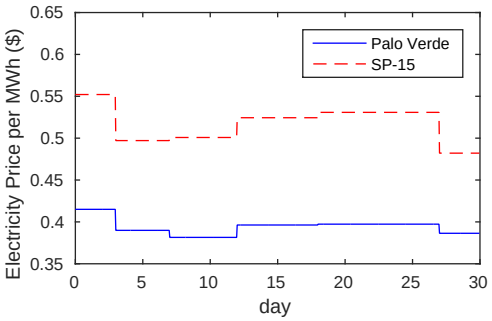


Fig. 8. Wholesale electricity market price for the hubs.

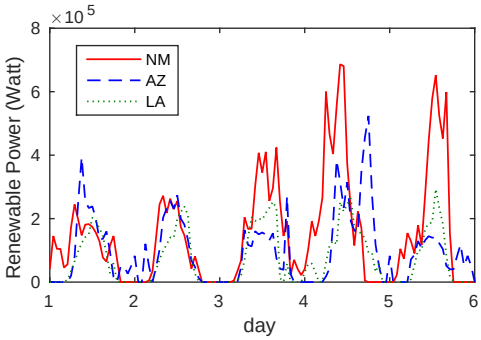


Fig. 9. Renewable power generation for five days.

of meteorological data traces in the database of *National Renewable Energy Laboratory* (NREL) [9]. All data centers use wind turbines (GE 1.5MW wind turbine with efficiency of 40%) and solar panels ($500m^2$ with efficiency of 30%) to produce electricity for local usage. Capacity of all data centers are set in a way that it does not lead to any request rejections. As shown in Figure 7, data centers are assumed to be connected to the utility grid and are able to buy electricity from a nearby hub (delivery points). The wholesale electricity price information for each hub is collected from the website of *Energy Information Administration* (EIA) [10]. Data center sites in NM and AZ are connected to *Palo Verde* hub and the data center site in LA is connected to the *SP-15* hub. Figure 8 shows the price of electricity for the wholesale market at each hub for a period of 30 days.

3) Renewable traces: To capture the availability of solar and wind energy in the location of each data center, we use meteorological data traces by NREL [9] with 1-hour granularity between 1st and 29th of May 2013.

We presume each data center uses a GE 1.5MW wind turbine to generate wind power. To estimate the average wind power production per hour, the model proposed by Fripp and Wiser [11] is employed where the wind speed, the air temperature, and the air pressure measurements in the location of each data center collected from NREL traces are fed into the model.

Similarly, the Global Horizontal Irradiance (GHI) in the location of each data center is used to calculate the output for solar photovoltaics (PV) power. Each data center uses power generated by the PV panels of $500m^2$ total area with tilt angle of 45° degree and PV cell efficiency of 30%. We calculate the PV power module output on the tilt surface based on the model in [12].

The summation of the power generated from these two sources per hour is computed as hourly available renewable power for every data centers. Figure 9 shows a sample five days of the total generated power from the renewable sources for different data centers.

4) Benchmark Algorithms: In Section III, we showed the exponential time complexity of the offline optimal GLB algorithm and its intractability. Running the simulation of such an algorithm is prohibitive in the settings of the discussed case study. Therefore, we consider the following benchmarks GLB algorithms in order to study the performance of FLB.

Future-Aware Best Fit (FABEF): This algorithm is similar to Algorithm 1 without performing recursive calculations. That is, it routes each request to a data center leading to the lowest cost of the request accommodation within the future window irrespective of upcoming requests. The FABEF algorithm has the full knowledge of electricity price and renewable energy traces to a certain future time window.

Round Robin (RR) : It routes requests to data centers in circular order with null information about the status of data centers.

Highest Available Renewable First (HAREF): The HAREF algorithm routes requests to the data center with the highest amount of available renewable energy at the current time slot.

B. Experimental Results and Analysis

We utilize CloudSim [13], a discrete-event Cloud simulator including models of virtualized computing infrastructures, and create a supporting module for the GLB. We model a cloud provider with a set of three data centers (zones) configured and connected to a nearby hub as explained in subsection V-A2. We use the electricity price for each hub form EIA [10] in the period of experiment. We refer to each data center according to its state, namely *California* (CA), *Arizona* (AZ), and *New Mexico* (NM). The renewable (green) power availability at each data center is also defined according to the settings in subsection V-A3. We performed several numerical experiments to evaluate the cost and green power utilization of FLB. Each experiment carried out for 29 days according to the workload setup driven form real-world traces of Google as explained in Section V-A1.

We set the cost per each data center as total units of power used from the utility grid multiplied by the corresponding electricity price. Total cost of the cloud provider is calculated as the summation of all data centers' cost. The green energy utilization is also computed as the ratio of total number green power units used to the total units of green power generated in each data center for the period of experiment.

1) The performance analysis of FLB against benchmark algorithms: In this section, we study the cost savings and renewable power utilization of FLB algorithm against other benchmark algorithms. All reported results are normalized to the outcome of RR algorithm which has the worst performance and both FLB and FABEF use the window size of 5 hours. Figure 10 shows the normalized total cost divided by the cost in each data center. The figure illustrates that FABEF, which is

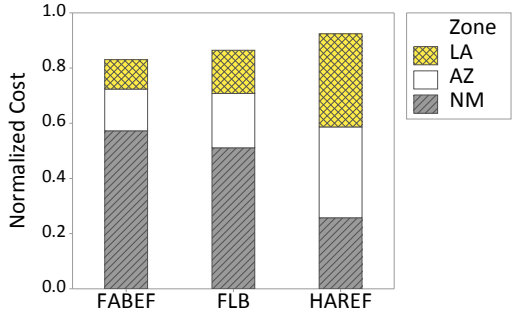


Fig. 10. The total cost of different algorithms normalized to the outcome of the RR algorithm.

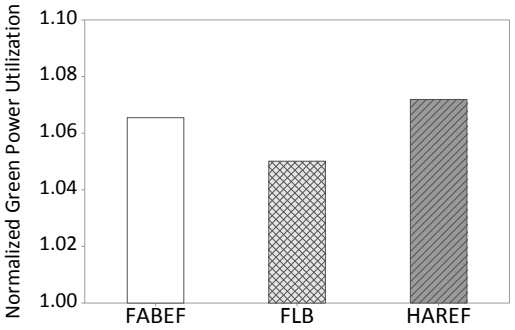


Fig. 11. The green power utilization of different algorithms normalized to the outcome of the RR algorithm.

equipped with future knowledge, has the best performance with 14% cost savings against simple RR algorithm. FLB, which uses past information for the decision making, has the second best results with only 2.5% lower cost savings compared to FABEF. The HAREF algorithm is able to gain 7.5% cost savings which shows that cloud providers are able to save cost even using a simple HAREF strategy compared to naïve algorithm such as RR.

Figure 11 depicts the total amount of green power utilization by each algorithm. As it can be seen, HAREF, which greedily routes requests to a data center with highest currently available green power, utilizes the highest amount of green energy. However, this does not necessarily lead to the lowest cost for the provider. As shown in Figure 10, both FABEF and FLB generate significantly lower cost compared to HAREF while they utilize green energy 0.7% and 2% lower, respectively (Figure 11).

2) Impact of recent history window size on the FLB performance: We further analyze the cost performance of FLB under different sizes of the recent history window. Figure 12 shows that the normalized total cost of FLB and FABEF as respectively past and future window size is increased from 1 hour to 12 hours. As we expected, the cost performance of FABEF increases when more future data is available to the algorithm. However, increasing the size of the recent past history for FLB algorithm does not necessarily enhance its performance. As it can be seen, in our case study, the best cost performance is achieved when the window size of 5 hours is selected. In practice, in order to get the best outcome of FLB, one must tune the window size according to the settings of

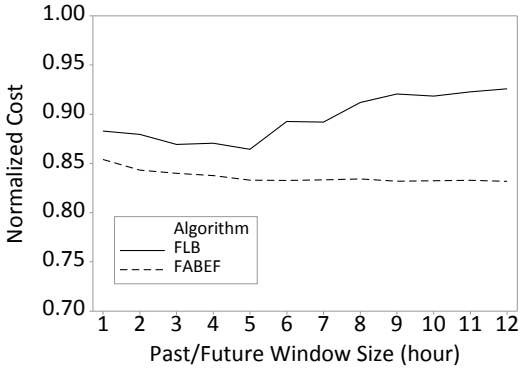


Fig. 12. Effect of window size on the total cost performance of FLB and FABEF normalized to the outcome of the RR algorithm.

workload and the environment.

VI. Related Work

Over the last decade, power management techniques to minimize data centers' costs and environmental impacts have gained considerable attention by both academia and industry. Large data centers such as those used by big companies like Google and Amazon can host thousands of physical servers and require up to tens of megawatts electricity to power them [14]. As result, service providers are under huge pressure to reduce their energy consumption and its associated costs.

Most of the early research studies on energy efficiency focus on making green data centers using optimization techniques within a single data center; techniques such as CPU dynamic voltage and frequency scaling (DVFS) [15], virtualization and VM consolidation [16], [17], and workload scheduling [18]. An extensive survey and taxonomy of these can be found in [19]. A large body of recent literature focus on reducing energy costs targeting geographically distributed data centers. These group of works mainly devise techniques for workload distribution across Geo-distributed data centers in order to achieve performance objectives such as cost minimization, renewable energy maximization, and emission minimization. Rahman et al. [20] present a comprehensive survey on data center power management using geographic load balancing.

One of the early studies on GLB is done by Liu et al. [3]. Using GLB, they propose algorithms to maximize renewable energy utilization and show how dynamic electricity price can affect brown energy usage. An extension to this work has been done by Lin et al. [2], where they propose online algorithms to exploit the potential of geographical diversity of internet-scale services on renewable energy utilization. As part of their research, they show the optimal portfolio of solar and wind energy sources in GLB. Similar to their work, we consider GLB to reduce energy cost and to maximize renewable energy utilization. However, we propose fuzzy logic-based load balancing algorithm with no future knowledge, while they propose online algorithms optimizing over a window of predicted future loads.

Similarly, He et al. [21] considered the sustainability of data centers by proposing socially-responsible load scheduling for data centers where they consider emission cost as the social

cost. Chen et al. [22] proposed a scheduling algorithm that considers the workload fluctuation, jobs' deadline, variable green energy supply, outside temperature, and data center cooling dynamics. All these studies consider data centers with on-site free of charge power generations from renewable energy sources.

There is another class of research efforts assume that data center must pay for the power drawn from the off-site renewable energy source. Le et al. [23] considered load distribution across data center sites by including capping energy usage from non-renewable sources. Gao et al. [24] propose a framework, which is compared to the method by Le et al. [23], for request-routing and traffic engineering considering changes in workload and carbon footprint. They attempt to balance the three-way trade-off between access latency, carbon footprint, and electricity costs. Zhang et al. [25] proposed the GreenWare framework to maximize the renewable energy utilization of Geo-distributed data centers. They assume that renewable energy is more expensive than brown energy.

VII. Conclusions and Future Work

We proposed a fuzzy logic-based algorithm for cost and energy efficient load balancing among multiple data centers of a cloud service provider. We also showed that optimal offline geographical load balancing is hard to achieve even if the perfect future knowledge about upcoming requests and their characteristics (i.e., arrival, size and lifetime), availability of renewable energy sources, and electricity price are known in advance. Considering the complexity of optimal load balancing and the difficulty of achieving complete and exact future knowledge, we designed and proposed an algorithm based on the fuzzy logic inference systems called FLB. The FLB algorithm provides a non-linear mapping from the inputs like recent utilization of the renewable power, the amount of brown (utility grid) energy consumption, and average electricity price to an output showing the appropriateness of the data center for the request redirection. Our evaluation using simulation of a case study designed according to real-world traces of workload, renewable energy sources, and electricity market prices showed that our proposed method is able to significantly reduce the cost of the cloud provider. We also demonstrated the impact of recent history window size on the performance of the FLB algorithm and its importance for the decision making.

References

[1] NRDC and Anthesis, "Scaling up energy efficiency across the data center industry: Evaluating key drivers and barriers," Natural Resources Defense Council, Tech. Rep., 2014.

[2] M. Lin, Z. Liu, A. Wierman, and L. Andrew, "Online algorithms for geographical load balancing," in *Proceedings of the International Green Computing Conference (IGCC '12)*, June 2012, pp. 1–10.

[3] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening geographical load balancing," in *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '11)*. New York, NY, USA: ACM, 2011, pp. 233–244.

[4] M. Adnan, R. Sugihara, and R. Gupta, "Energy efficient geographical load balancing via dynamic deferral of workload," in *Proceedings of 5th IEEE International Conference on Cloud Computing (CLOUD '12)*, June 2012, pp. 188–195.

[5] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007, vol. 703.

[6] E. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1 – 13, 1975.

[7] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format + schema," Google Inc., Mountain View, CA, USA, Technical Report, Nov 2011, [Online]. Available: http://code.google.com/p/googleclusterdata/wiki/TraceVersion2.

[8] S. Bergqvist and H. Geng, *Data Center Handbook*. Wiley, 2014, ch. Using Direct Current Network in Data Centers, pp. 523–532.

[9] "Measurement and Instrumentation Data Center (MIDC)," http://www.nrel.gov/midc/.

[10] "Wholesale Electricity and Natural Gas Market Data," http://www.eia.gov/electricity/wholesale/.

[11] M. Fripp and R. H. Wiser, "Effects of temporal wind patterns on the value of wind-generated electricity in california and the northwest," *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 477–485, 2008.

[12] "Solar Radiation on a Tilted Surface," http://pveducation.org/pvcdrom/properties-of-sunlight/solar-radiation-on-tilted-surface.

[13] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, Jan. 2011.

[14] F. Kong and X. Liu, "A survey on green-energy-aware power management for datacenters," *ACM Computing Surveys*, vol. 47, no. 2, pp. 30:1–30:38, Nov. 2014.

[15] C.-M. Wu, R.-S. Chang, and H.-Y. Chan, "A green energy-efficient scheduling algorithm using the dvfs technique for cloud datacenters," *Future Generation Computer Systems*, vol. 37, pp. 141–147, 2014.

[16] A. Beloglazov and R. Buyya, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1366–1379, July 2013.

[17] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," in *Proceedings of the USENIX 2008 conference on Power aware computing and systems (HotPower '08)*, vol. 10, San Diego, California, 2008.

[18] M. Ghamkhari and H. Mohsenian-Rad, "Energy and performance management of green data centers: A profit maximization approach," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 1017–1025, June 2013.

[19] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Advances in computers*, vol. 82, no. 2, pp. 47–111, 2011.

[20] A. Rahman, X. Liu, and F. Kong, "A survey on geographic load balancing based data center power management in the smart grid environment," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 214–233, Jan. 2014.

[21] J. He, X. Deng, D. Wu, Y. Wen, and D. Wu, "Socially-responsible load scheduling algorithms for sustainable data centers over smart grid," in *Proceedings of Third IEEE International Conference on Smart Grid Communications (SmartGridComm '12)*, Nov 2012, pp. 406–411.

[22] C. Chen, B. He, and X. Tang, "Green-aware workload scheduling in geographically distributed data centers," in *Proceedings of 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom '12)*, Dec 2012, pp. 82–89.

[23] K. Le, O. Bilgir, R. Bianchini, M. Martonosi, and T. D. Nguyen, "Managing the cost, energy consumption, and carbon footprint of internet services," in *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '10)*. New York, NY, USA: ACM, 2010, pp. 357–358.

[24] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav, "It's not easy being green," *SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 211–222, Aug. 2012.

[25] Y. Zhang, Y. Wang, and X. Wang, "Greenware: Greening cloud-scale data centers to maximize the use of renewable energy," in *Middleware 2011*, ser. Lecture Notes in Computer Science, F. Kon and A.-M. Kermarrec, Eds. Springer Berlin Heidelberg, 2011, vol. 7049, pp. 143–164.