Geographical Load Balancing for Online Service Applications in Distributed Datacenters

Hadi Goudarzi and Massoud Pedram
University of Southern California
Department of Electrical Engineering - Systems
{*hgoudarz, pedram*}*@usc.edu*

Abstract — This work focuses on the load balancing problem for online service applications (which are response time-sensitive) considering a distributed cloud system comprised of geographically dispersed, heterogeneous datacenters. An offline solution based on force-directed scheduling is presented, which can determine the application placement for long periods of time. The solution is then extended to do online application placement and migration for geographically distributed datacenters based on predictions about the application lifetimes, workload intensities, dynamic energy prices, and renewable energy generation capacities at different datacenters in the cloud system. The simulation results demonstrate 27% to 40% improvement using the proposed algorithms with respect to the method that does not consider the geographical load balancing.

I. INTRODUCTION

Demand for computing power has been increasing due to the penetration of information technologies in our daily interactions with the world both at personal and communal levels, encompassing business, commerce, education, manufacturing, and communication services. Dramatic increase in the computing resources requires a scalable and dependable *information technology* (IT) infrastructure comprising of servers, storage, network bandwidth, physical infrastructure, electrical grid, personnel and billions of dollars in capital expenditure and operational cost to name a few.

Datacenters associated with a cloud system are typically geographically distributed, yet connected together with dedicated high-bandwidth communication links. This helps reduce the peak power demand of the datacenters on the local power grid, allows for more fault tolerant and reliable operation of the IT infrastructure, and even, lowers cost of ownership. A datacenter itself comprises thousands to tens of thousands of server machines, working in tandem to provide services to the clients, see for example [1] and [2]. These datacenters can be owned by one cloud provider or may be federated.

Datacenters are usually designed for the worst-case workload. At the same time, datacenter workload changes drastically depending on the time of the day and day of the week. Considering the dynamic energy pricing trend [3], price of the electrical energy purchased form the utility companies may be a function of time of day or the peak power consumed by the consumer. Energy prices at different sites of a geographically distributed cloud system can be different due to local time differences and differences in local utility company's energy prices. To reduce the reliance on brown sources of electricity and supplement/diversify the power generation sources for a datacenter, there is a trend to generate electricity from renewable sources such as wind and solar at the datacenters' site [4, 5]. Geographically distributed datacenters associated with a cloud system create load balancing opportunities that can result in a reduction in the total number of computing resources provisioned in datacenters (considering the time difference between peak workload times in different locations), as well as lowering of the operational cost of each datacenter by purchasing electrical energy at lower prices (considering dynamic energy prices at each site depending on the local time) and/or increasing the portion of the renewable power generated in some datacenters.

Geographical load balancing (GLB) can be defined as a series of decisions about online assignment and/or migration of *virtual machines* (VMs) or computational tasks to geographically distributed datacenters in order to meet the *service level agreements* (SLAs) or service deadlines for VMs/tasks and to decrease the operational cost of the cloud system.

Effectiveness of the GLB in case of offline computation assignment and scheduling has been demonstrated in previous work [6, 7]. Most of the previous work that has focused on the GLB problem for online service applications, e.g., [8, 9, 10], simplify the VM placement and migration problem to a request forwarding problem for a VM type or a collection of VMs. This representation ignores the heterogeneity of VMs, the VM packing problem, and real VM migration cost and can thus result in low performance in a real cloud system.

In this work, we focus on the GLB problem for heterogeneous online service applications that are response time-sensitive. Communication latency, queuing and service delays, and VM migration penalty are the most important factors for determining the VM to datacenter assignment solution. The availability of each type of resource in a datacenter, peak power capacity, and varying *power usage effectiveness* (PUE) of a datacenter are considered in modeling the datacenter. There are two versions of the GLB solution: (i) an offline solution, which considers every optimization variable to be determined deterministically in order to derive a complete VM placement and migration solution for a long period of time, and (ii) an online solution, which uses prediction of the variables for the future to derive VM placement and migration for a short period of time. The offline solution can be used during the design of geographically distributed datacenters to

*This research is sponsored in part by a grant from the Division of Computer and Communication Foundations of the National Science Foundation.

reduce the initial capital expenditure and expected operational cost of the datacenter.

This paper presents a novel algorithm based on force-directed scheduling [11] to solve the offline problem for geographically distributed datacenters. This algorithm is subsequently extended to an online solution to perform periodic VM placement and migration management for online service applications based on the prediction of application active periods, workload types and intensities, electrical energy prices, and potential generation of renewable energy in the near future. The effectiveness of the proposed solutions is demonstrated by comparison them to a case without the GLB capability.

This paper is organized as follows. The most relevant prior work is reviewed in section II. Parameter definition and precise problem formulation for the offline scenario are given in sections III and IV. The offline version of the solution is presented in section V while online problem formulation and solution are presented in section VI. Simulation results are presented in section VII and paper is concluded at section 0.

II. Related work

The GLB can be seen as the high-level resource management problem in the cloud system. Resource management problems in cloud computing systems have attracted a lot of attention in recent years. Datacenter, VM and SLA modeling, and resource management solutions inside a datacenter are extensively discussed in the previous work, cf. [12, 13, 14, 15, 16, 17, 18]. In this section, a review of the most relevant work to the GLB problem is provided.

Some prior work has focused on reducing the operational cost of the cloud system by using the load balancing opportunity – see [19] and [20]. Model predictive control has been used to solve the GLB problem using the estimated future load, e.g., [21] and [22]. These studies consider homogenous datacenters (where all servers are identical), which is far from the real-world situations. Reference [8] considers heterogeneous datacenters (comprised of servers with different performance and power dissipation figures, and even with instruction sets), which results in a more elaborate load balancing mechanism. Unfortunately, this work still ignores the heterogeneity of VMs, VM packing problem, and realistic VM migration cost and can result in low performance under realistic scenarios.

GLB increases the chances for effective utilization of renewable power sources in datacenters. For instance, a recent work in [23] investigates the feasibility of powering up a geographically distributed datacenter with only renewable power. A possible disadvantage of GLB is that the access to cheap electrical energy purchased from the local utility companies may result in an increase in the datacenter's power consumption. Considering the environmental cost of energy usage (e.g., carbon emission) can eliminate this possibility. For example, reference [9] shows that if the electricity price is set based on the share of the brown energy to the total produced energy, GLB can reduce the brown energy usage. Similarly, Le et al. [10] present algorithms that reduce the brown energy usage in geographical distributed datacenters.

Considering offline computation adds another perspective to the GLB problem i.e., the possibility of computation deferral. Computation deferral is only appropriate for batch applications with loose deadlines and can be used in combination with online service application load scheduling to further reduce the total energy cost or brown energy consumption of a datacenter. Reference [6] focuses on computation scheduling in datacenters and computation deferral to minimize the energy cost. Reference [7] solves the GLB problem considering online service and batch applications and cooling supply selection in datacenters. The cooling supply choices considered in this paper are to use a chiller or outside air cooling.

III. Parameter definitions

The GLB solution for online service applications is a periodic VM assignment to and/or VM migration across geographically distributed datacenters if necessary. The objective of the GLB problem is to minimize the operational cost (the electrical energy bill plus SLA penalty) of the cloud system while satisfying resource, peak power capacity, and SLA constraints. Note that the decisions in the GLB solution are focused on the cloud-level VM assignment and migration. Each datacenter has its own VM management that assigns VMs to its servers and migrates VMs. The datacenter-level VM management and migration are out of the scope of this paper. An exemplary figure for a geographically distributed datacenter is shown in Figure 1.

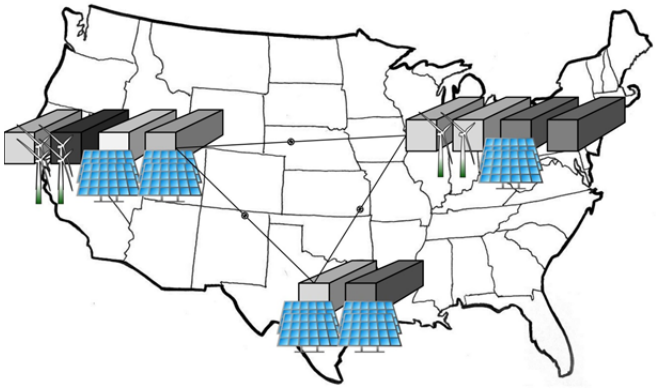

Figure 1 – An exemplary figure for a geographically distributed cloud system

In this work we focus on solving the GLB problem in case of heterogeneous VMs and heterogeneous servers in each datacenter. We present two versions of the solution to this problem: (i) An offline solution considering predicted workload, known renewable power generation capability, and dynamic electrical electricity prices; (ii) A periodic online solution to decide about the VM placement and migration for the current time based on the immediate information and predictions about the future. Note that the assumption in the offline version of the problem is simplistic but the offline solution can be used in capacity provisioning for datacenters, determining datacenter site locations, or the amount of renewable power source construction near each datacenter. Moreover, as shown in this paper, the offline solution can be extended to the online version, which can be used in VM management in a cloud system.

The time axis in the GLB problem is divided into time slots called *epochs*. Each epoch is identified by a unique id, denoted by τ. T_e denotes the duration of each epoch, which is in the order of a few minutes to as much as an hour. New VMs are only admitted at the beginning of each epoch. Similarly, decisions about VM migration and placement are only applied at the beginning of each epoch. In addition to this decision making

process, a reactive manager migrates VMs between different datacenters in the case of drastic workload changes, which may create SLA, peak power, or thermal emergencies.

The solution to the GLB problem involves information about (or prediction of) the dynamic energy prices, renewable power generation capability, VM workload, and VM active period. The quality of these predictions determines the quality of the proposed solution to the GLB problem. In the first part of the paper, we consider an offline version of the problem that assumes perfect prediction of these parameters and determines the complete VM placement and migration for a long period of time e.g., a full day. Definitions of parameters for the offline version of the problem are given next.

T denotes the set of consecutive epochs that we consider for the offline version of the GLB. Similar to the online solution, the offline solution changes the VM assignment solution only at the beginning of each epoch. Each VM and each datacenter are identified by a unique id, denoted by i and d respectively. $I(\tau)$ denotes the set of active VMs in each epoch and D denotes the set of geographically distributed datacenters.

A *time-of-use* (TOU) dependent energy pricing scheme is considered for each utility company. The energy price is assumed to be fixed for each epoch. $EP^d(\tau)$ denote the energy price in datacenter d during epoch τ. TOU-dependent energy pricing scheme (in contrast to peak-power dependent energy pricing) enables one to ignore the time variation of renewable power generated in local renewable power facilities during an epoch and model the amount of generated renewable power by the average generated power in that epoch, which is denoted by $G^d(\tau)$. The allowed peak power consumption of a datacenter is determined by the power delivery network in the datacenter and is denoted by $P^{d,max}$. To translate the average power consumption to $P^{d,max}$, *peak to average power ratio* ($PAR^d(\tau)$) is used. This parameter depends on the resource capacity of the datacenter and the set of VMs assigned to the datacenter.

The PUE factor of a datacenter, which is defined as the ratio between total power consumption of the datacenter to the power consumed by the IT equipment in the datacenter, is dependent on the datacenter design (including facility planning and management and cooling technology) and the amount of instantaneous power consumption. We consider the PUE factor to be decomposed to a constant factor (Eff^d), which accounts for the *uninterrupted power supply* (UPS) inefficiencies within the datacenter, and a load-dependent factor ($1 + 1/COP^d(\tau)$), which captures the inefficiency of the air conditioning units in the datacenter. In the load-dependent factor, the *coefficient-of-performance* ($COP^d(\tau)$), which models the amount of power consumed by the air conditioning units, depends on the temperature of the supplied cold air, which is in turn a function of the IT equipment power dissipation in the datacenter. Optimal $COP^d(\tau)$ is a monotonically decreasing function of the average power consumption in the datacenter.

We consider only the processing capacity as the resource in each datacenter (consideration of other resource types such as the storage or network bandwidth falls outside the scope of present paper). To model each datacenter more accurately, we consider datacenters with heterogeneous servers. Each server type is identified by a unique id j in each datacenter and the set of server types in each datacenter is shown by J^d. $C^{d,j}$ denotes the number of servers of type j in datacenter d. Different server types have different characteristics in terms of their processing speed (CPU cycles per second) and power consumption.

Due to non-energy proportional behavior of the servers [24], it is important to translate the amount of resources required in the server pool to the number of active servers. To capture the VM packing effect, we assume that any active server of type j, is utilized by an average value (smaller than one, e.g., 0.8) denoted by $\bar{\phi}^j$. The rationale is that considering any resource requirement value, server-level power management strategies including server consolidation or dynamic voltage and frequency scaling methods are employed in the datacenter ensuring that an active server is utilized at a high level so that we avoid having to pay the penalty associated with the non-energy proportionality behavior of the servers. This average utilization level for different server types may not be the same because the characteristics and configuration of each server type in terms of its power consumption vs. utilization level curve as well as the amount of memory, local disk size, network interface bandwidth are generally different.

The average power consumption of each of these resource types in datacenter can be found by multiplying the average power consumption of a typically utilized server of given type ($\bar{\phi}^j P_j^p + P_j^0$) by the number of servers needed to support the assigned VMs in the datacenter. In this formula, P_j^0 and P_j^p denote the idle and utilization-dependent power consumption of a server of type j.

Each client of the target cloud system creates one VM to execute its application. The SLA for online service application determines a target response time for requests generated by the VM. The cloud provider must guarantee the satisfaction of this response time constraint for a percentage of incoming requests (e.g. 95%) and agrees to pay a fixed penalty for any request violating the response time constraint. Moreover, SLAs determine VM migration cost, which is the penalty for service outage due to VM migration. $mc_i^{d,d'}(\tau)$ denotes the VM migration cost between datacenter d and d'.

Let $x_i^{d,j}(\tau)$ denote the amount of servers of type j in datacenter d allocated to VM i in epoch τ. To determine this resource allocation parameter for each VM, a performance model must be considered. Each VM will have different resource requirements and exhibit different response time behavior if it is assigned to different server types. Moreover, dependence of a VM's request response time in a host datacenter can be determined based on the communication distance between the VM's origination point and the host datacenter, the data rate in dedicated communication channels, the packet size of the incoming requests and outgoing response.

Performance models proposed in the literature can help translate the resource allocation parameter to specific SLA violation cost or price based on the client's SLA requirements, VM workload in the epoch, execution behavior of VM on the specific server type, and the communication latency. The performance model can be abstracted by parameter $r_i^{d,j}(\tau)$ and function $f(x_i^{d,j}(\tau))$ that denote the least amount of the computing resource needed in order to guarantee satisfaction of the SLA constraint and the expected SLA cost of VM i in epoch τ with allocation parameter $x_i^{d,j}(\tau)$, respectively. According to definition, $f(0) = 0$. Ignoring $x_i^{d,j}(\tau) = 0$, this function is

monotonically decreasing with respect to $x_i^{d,j}(\tau)$. If the communication latency of assigning a VM to a datacenter violates the SLA response time constraint, parameter $r_i^{d,j}(\tau)$ will be equal to infinity in order to avoid such assignments.

Note that constraint $x_i^{d,j}(\tau) \geq r_i^{d,j}(\tau)$ guarantees that the SLA constraint will be satisfied based on the assumed performance model but in order to satisfy the SLA constraints, the host server monitors the performance of the application and in case of SLA violation increases the amount of resource allocated to it or requests VM migration from the datacenter-level resource manager.

IV. Problem Formulation for the Offline Problem

The offline version of the GLB problem can be formulated as follows:

$$Min \sum_{\tau \in \mathrm{T}} T_e \sum_{d \in D} EP^d(\tau)\left(P^d(\tau) - G^d(\tau)\right)^+ + \sum_{\tau \in \mathrm{T}} \sum_{i \in I(\tau)} \left(mc_i(\tau) + \sum_{d \in D} \sum_{j \in J^d} f(x_i^{d,j}(\tau)) \right)$$

subject to:

$$P^d(\tau) = \frac{1}{Eff^d}\left(1 + \frac{1}{COP^d(\tau)}\right) \sum_{j \in J^d} \left(\left(\bar{\phi}^j P_j^p + P_j^0\right) \times \sum_{i \in I(\tau)} x_i^{d,j}(\tau)/\bar{\phi}^j \right) \quad (1)$$

$$\sum_{d \in D} y_i^d(\tau) = 1 \quad \forall i \in I(\tau)\ ,\ y_i^d(\tau) \geq 0\ ,\ y_i^d(\tau) \leq sign\left(\sum_{j \in J^d} x_i^{d,j}(\tau)\right)\ ,\ y_i^d(\tau) \in \{0,1\} \quad (2)$$

$$1 \geq x_i^{d,j}(\tau) \geq 0, \quad \sum_{d \in D} \sum_{j \in J^d} \left(x_i^{d,j}(\tau)/r_i^{d,j}(\tau) - 1 + \epsilon\right)^+ > 0 \quad \forall i \in I(\tau) \quad (3)$$

$$\sum_{i \in I(\tau)} x_i^{d,j}(\tau) \leq C^{d,j} \quad \forall d \in D\ \&\ \forall j \in J^d \quad (4)$$

$$mc_i(\tau) \geq mc_i^{d,d'} y_i^d(\tau) y_i^{d'}(\tau - 1) \quad \forall d, d' \in D \quad (5)$$

$$P^d(\tau) PAR^d(\tau) \leq P^{d,max} \quad (6)$$

where $(A)^+$ denotes the $\max(A, 0)$ and Parameter ϵ is a very small positive value. Note that $sign(0)$ is equal to 0.

The optimization parameters in this problem include the assignment parameters ($y_i^d(\tau)$) and the allocation parameters ($x_i^{d,j}(\tau)$). The objective function includes three terms: (i) the energy cost paid to the utility companies, (ii) the VM migration cost, and, (iii) the SLA cost of VMs based on the VM assignment and amount of resources allocated to them.

Equation (1) determines the average power consumption of each datacenter based on the allocated resource to VMs. Constraint (2) determines the pseudo-Boolean assignment parameter for each VM in each epoch. Constraint (3) forces the amount of resources allocated to each VM to be greater than $r_i^{d,j}(\tau)$. Resource capacity constraint for each server type in each datacenter is captured by constraint (4). Constraint (5) determines the migration cost associated with each VM. The migration cost is equal to zero unless the VM is migrated from datacenter d' to d in epoch τ. In the latter case, the migration cost is equal to $mc_i^{d',d}$. In order to consider the initial VM assignment solution, if VM i is initially assigned to datacenter d, $y_i^d(-1)$ is set to one. Finally, constraint (6) captures the peak power capacity constraint in each datacenter.

The GLB problem is an NP-hard optimization problem. Most of the previous work [8, 9, 10, 19] has focused on solving the GLB problem with continuous workload approximation. The problem can subsequently be solved using convex optimization methods. The continuous approximation of the GLB problem is acceptable in case of homogenous VMs or simple request forwarding scenarios in a cloud system. This simplification cannot, however, accurately capture the VM migration cost and may result in poor performance due to the necessity of deciding about the actual VM placement after finalizing the load balancing solution. In this work, we present an online and offline solution to the GLB problem for online service applications in the cloud system.

V. Algorithm for the Offline Solution

As explained in section III, in the offline version of the problem, we assume that every input parameter is known as opposed to an online scenario in which these parameters are only predicted with certain confidence. The input parameters in this problem are the VM arrival time and active period, the VM workload in each epoch, energy price and generated power in the renewable power plant for each datacenter. We consider these parameters to be fixed during an epoch. Making this assumption means that the frequency of drastic changes in the system is considered to be greater than the frequency of applying the optimization solution.

The GLB problem involves a resource allocation problem for VMs assigned to each datacenter at each epoch. To determine the optimal amount of resources that need to be allocated to a VM to minimize the summation of energy and SLA costs, we need to know the effective energy price and the PUE of a datacenter. It is obvious that these values cannot be determined without knowing the average power consumption in the target epoch but we can estimate $COP^d(\tau)$ and $P^d(\tau)$ by using their typical values in previous epochs with similar conditions. The problem of finding the best resource allocation parameter for VM i if it is assigned to server type j in datacenter d may be formulated as follows:

$$Min\ EC^d(\tau) \frac{\left(\hat{P}^d(\tau) - G^d(\tau)\right)^+}{\hat{P}^d(\tau)} \frac{1}{Eff^d}\left(1 + \frac{1}{\widehat{COP}^d(\tau)}\right) \times T_e(\bar{\phi}^j P_j^p + P_j^0)\frac{x_i^{d,j}(\tau)}{\bar{\phi}^j} + f(x_i^{d,j}(\tau))$$

subject to:

$$x_i^{d,j}(\tau) \geq r_i^{d,j}(\tau) \quad (7)$$

In this formulation, $\hat{P}^d(\tau)$ and $\widehat{COP}^d(\tau)$ denote the estimated average power consumption and COP, respectively. Considering a non-increasing SLA cost function, the problem has only one solution in which $x_i^{d,j}(\tau) = r_i^{d,j}(\tau)$, $x_i^{d,j}(\tau) = 1$ or it satisfies the following equality (KKT conditions):

$$\frac{\partial f\left(x_i^{d,j}(\tau)\right)}{\partial x_i^{d,j}(\tau)} = -EC^d(\tau) \frac{\left(\hat{P}^d(\tau) - G^d(\tau)\right)^+}{\hat{P}^d(\tau)} \times \frac{1}{Eff^d}\left(1 + \frac{1}{\widehat{COP}^d(\tau)}\right) T_e\left(P_j^p + \frac{P_j^0}{\bar{\phi}^j}\right) \quad (8)$$

Considering a constant communication delay for assigning a VM to a datacenter, a closed form solution can be found for (8) by using the M/M/1 queuing model, cf. [16]. In case of more

complicated SLAs or queuing models, it may not be possible to obtain a closed form solution for this problem, but a numerical solution can be used in such cases. In the rest of this paper, $x_i^{d,j}(\tau)$ denotes the solution of (8) or zero depending on the value of $y_i^d(\tau)$. Note that, at any point of the algorithm where all VMs are assigned to a datacenter for an epoch, the value of $x_i^{d,j}(\tau)$ can be updated based on real values of $P^d(\tau)$ and $COP^d(\tau)$.

The GLB problem considering VMs with lifetimes greater than single epoch is more complicated than finding the best VM placement solution for each epoch because a VM may cost less if it is not assigned to its best datacenter in the current epoch so as to avoid having to pay for costly VM migration in a next epoch. To be able to find an efficient and high-performance solution for the GLB problem, we propose a force-directed load balancing (FDLB) algorithm, which determines VM placement solution based on force-directed scheduling (FDS) [11].

FDS is one of the notable scheduling techniques in high-level synthesis. It is a technique used to schedule directed acyclic task graphs so as to minimize the resource usage under a latency constraint. This technique maps the scheduling problem to the problem of minimizing forces in a physical system which is subsequently solved by iteratively reducing the total force by task movements between time slots. In reference [25], we applied this technique to the task scheduling in demand response problem.

To solve the GLB problem using the FDS technique, $|T|$ instances of each datacenter (one for each epoch) and an instance of each VM for each epoch in its active period are created. Note that, the instance of a VM in epoch τ only has interactions with datacenter instances in that epoch and the VM instances in epoch $\tau-1$ and $\tau+1$ (if they exist). Forces in this system are defined based on different terms in the objective functions and resource and peak power capacities in datacenters. Assigning an instance of VM i in epoch τ to server type j in datacenter d creates the following force in the system:

$$Force_i^{d,j}(\tau) = FO_i^{d,j}(\tau) + FM_i(\tau) + FC_i^{d,j}(\tau) + FP_i^{d,j}(\tau) + FR_i^{d,j}(\tau) \quad (9)$$

where:

$$FO_i^{d,j}(\tau) = EP^d(\tau)\frac{\left(P^d(\tau)-G^d(\tau)\right)^+}{P^d(\tau)}T_e\left(1+\frac{1}{COP^d(\tau)}\right)\times \frac{1}{Eff^d}\left(P_j^p+\frac{P_j^0}{\bar{\phi}^j}\right)x_i^{d,j}(\tau)+f\left(x_i^{d,j}(\tau)\right) \quad (10)$$

$$FM_i(\tau) = \sum_{d'\in D} mc_i^{d,d'}\left(y_i^{d'}(\tau+1)+y_i^{d'}(\tau-1)\right) \quad (11)$$

$$FC_i^{d,j}(\tau) = \frac{-1}{COP^d(\tau)^2}\frac{\partial COP^d\left(P^d(\tau)\right)}{\partial P^d(\tau)}\frac{1}{Eff^d}T_eP^d(\tau)\times EP^d(\tau)\frac{\left(P^d(\tau)-G^d(\tau)\right)^+}{P^d(\tau)}\left(P_j^p+\frac{P_j^0}{\bar{\phi}^j}\right)x_i^{d,j}(\tau) \quad (12)$$

$$FP_i^{d,j}(\tau) = \left(1-e^{-PUE\left(P_j^p+P_j^0/\bar{\phi}^j\right)x_i^{d,j}(\tau)}\right)\times e^{\left(P^d(\tau)PAR^d(\tau)-P^{d,max}(\tau)\right)} \quad (13)$$

$$FR_i^{d,j}(\tau) = \left(1-e^{-x_i^{d,j}(\tau)}\right)e^{\left(\sum_{i\in I(\tau)}x_i^{d,j}(\tau)-C^{d,j}\right)} \quad (14)$$

It can be seen that different force elements are defined for different parts of the objective function or constraints in the GLB problem as explained next. $FO_i^{d,j}(\tau)$ captures the energy and SLA costs based on the amount of resources allocated to the VM. $FM_i(\tau)$ captures the VM migration cost whereas $FC_i^{d,j}(\tau)$ captures the energy cost of the cooling power consumption change due to the average power consumption change. $FP_i^{d,j}(\tau)$ and $FR_i^{d,j}(\tau)$ capture the pressure on the peak power and server type j resource capacity constraints in the datacenter. Note that $FP_i^{d,j}(\tau)$ and $FR_i^{d,j}(\tau)$ do not have corresponding cost meaning, but are added to the force calculation to make sure the capacity constraints are satisfied.

Finding a feasible solution to minimize the objective function is equivalent to minimizing the summation of forces applied to VM instances. Starting from any solution, we can identify the VM instance movements (from a server type in a datacenter to another server type in a datacenter) that results in reducing the force and execute these movements to reach a lower operational cost. The order of these movements affects the final results because executing a movement changes the forces applied to some other VM instances.

The initial solution has a significant impact on the quality of the final solution for the GLB problem. To be able to perform gradual VM movement to reduce the total force, we consider an initial solution in which, each VM instance is cloned and uniformly distributed between possible resource types in different datacenters related to the target epoch. Let $N_i(\tau)$ denote the number of instances for VM i in epoch τ. The amount of resources allocated to new VM instances is replaced by $x_i^{d,j}(\tau)/N_i(\tau)$ and force components are calculated based on this value. Note that the SLA cost for these VM instances should be calculated from $f(x_i^{d,j}(\tau))/N_i(\tau)$ while the migration cost-related force calculation should consider multiple VM instances in neighboring epochs with appropriate weights. More precisely, $FM_i(\tau)$ for an instance of the VM should be replaced by the following term:

$$FM_i(\tau) = \frac{1}{N_i(\tau)}\sum_{d'\in D} mc_i^{d,d'}\left(\frac{y_i^{d'}(\tau+1)}{N_i(\tau+1)}+\frac{y_i^{d'}(\tau-1)}{N_i(\tau-1)}\right) \quad (15)$$

Starting from the initial solution, we need to merge instances associated with each VM in each epoch to reduce the number of instances related to each VM to one for each epoch ($N_i(\tau)=1$). Speed of the instance merging affects the run-time of the algorithm and the overall quality of the solution. We select a three-stage merging approach in which first we reduce the number of instances for each VM in each epoch to 4 and then reduce the number of instances to 2, and finally, determine the VM placement. In each stage, the best merging action (least force increase) between different VMs and different epochs is selected and executed until there are no VMs with more than the target number of instances in each epoch. To calculate the best merging action and its associated force, the total force applied to VM instances is calculated and subtracted from the best total force if the instances are reduced to the target number of instances. Note that any VM instance movement results in changes in forces applied to VM instances associated with the datacenters in that epoch and its own VM instances in the neighboring epochs. These force changes are captured in equation (9) but to calculate the next best VM movement, the value of force for affected VM instances needs to be updated.

After finalizing the VM placement solution, in case of resource or peak power capacity constraint violation in datacenters, the VM instance movement must be continued until

a feasible solution is reached. In this stage, VM instances can select any destination resource type in a datacenter in the corresponding epoch in contrast to gradual VM instance merging, which was limited to select destination(s) between current VM instance hosts. In addition to this stage, even without any peak power capacity or resource constraint violations, the VM movement can be continued to further reduce the total cost with the restriction that no VM movement that results in any constraint violations should be tried.

Considering this algorithm, we can formulize the datacenter and renewable power plant design problem to minimize the capital expenditure and operational cost. The presented algorithm can also be modified and used in the online VM management in a cloud system comprised of geographically distributed datacenters. Details of this extension are given next.

VI. Problem Formulation and Proposed Solution for the Online Version of the Problem

VM placement in a cloud system comprised of geographically distributed datacenters is performed at the beginning of each epoch based on the prediction of the optimization parameters. The online solution for the GLB problem determines the VM placement solution for the current epoch (denoted by t in this section) with the consideration of future epochs. To make a decision about a VM placement, we need to consider its active period, its workload in the next epochs, other VMs in the system including existing VMs and new VMs that will enter the cloud in the next epochs, and energy price and renewable energy generation for next epochs.

The cloud system cost ($CC(t)$) in epoch t can be formulated as follows:

$$CC(t) = \sum_{d\in D} EP^d(t)T_e\left(P^d(t) - G^d(t)\right)^+ + \sum_{i\in I(t)}\left(mc_i(t) + \sum_{d\in D}\sum_{j\in J^d} f\left(x_i^{d,j}(t)\right)\right) \quad (16)$$

The online version of the GLB problem tries to minimize the summation of $CC(t)$ and the costs of the future epochs ($CC(t+\tau)$) by VM placement for the current set of VMs.

The online GLB solution directly affects $CC(t)$ but only indirectly affects $CC(t+\tau)$. In contrast to straightforward calculation of $CC(t)$ based on the VM placement solution (considering perfect information about optimization parameters in epoch t), estimating $CC(t+\tau)$ is a difficult task due to the following missing information about epoch $t+\tau$:

(i) Existence of VM i ($i \in I(t)$) in epoch $t+\tau$. A probability value denoted by $pr_i(t+\tau)$ is considered to determine the probability of the VM to be active in epoch $t+\tau$. This probability is a decreasing function of τ.

(ii) VM i Workload ($i \in I(t)$) in epoch $t+\tau$. Considering the SLA, workload in our problem formulation may be translated into resource allocation parameters. Therefore, we consider predicted resource allocation parameters in epoch $t+\tau$, denoted by $\hat{x}_i^{d,j}(t+\tau)$.

(iii) Energy price and average renewable power generation. We consider $\widehat{EP}^d(t+\tau)$ and $\hat{G}^d(t+\tau)$ to represent the predicted energy price and average renewable power generation in epoch $t+\tau$.

(iv) The rest of active VMs ($I(t+\tau) - I(t)$) and their workload in that epoch. Instead of predicting a number of active VMs for epoch $t+\tau$, resource utilization related to those VMs in datacenters can be used. These resource utilization parameters can be found based on the state of the datacenters in similar scenarios (same energy price, renewable power generation and workload) after removing the resource utilization related to VMs that existed in epoch t. The amount of predicted *background* utilized resources for resource type j in datacenter d in epoch $t+\tau$ is denoted by $\hat{C}^{d,j}(t+\tau)$.

Parameters $pr_i(t+\tau)$ and $\hat{x}_i^{d,j}(t+\tau)$ can be estimated based on the historical data about the VM type and VM's original location. Energy price can be predicted based on the historical data or information received from utility companies and average renewable energy generation can be estimated based on weather prediction.

Based on the predicted optimization parameters, the online VM placement problem in a geographically distributed datacenter can be set up similar to the offline problem. A maximum application lifetime is considered for every VM and SLA and migration cost and allocation parameters for VM i in epoch $t+\tau$ are dampened by probability $pr_i(t+\tau)$. To simplify the formulation, in the following formulation, we consider the predicted parameters to be equal to their actual values for epoch t and set $pr_i(t) = 1$. The online GLB problem can thus be formulated as follows:

$$Min \sum_{\tau\in \mathrm{N}^0} T_e \sum_d \widehat{EP}^d(t+\tau)\left(P^d(t+\tau) - \hat{G}^d(t+\tau)\right)^+ + \sum_{\tau\in \mathrm{N}^0} pr_i(\tau) \sum_{i\in I(t)}\left(mc_i(t+\tau) + \sum_d\sum_j f\left(\hat{x}_i^{d,j}(t+\tau)\right)\right)$$

subject to constraints (2), (3), (5), (6) and

$$\sum_{i\in I(\tau)} pr_i(t+\tau)\hat{x}_i^{d,j}(t+\tau) \le C^{d,j} - \hat{C}^{d,j}(t+\tau) \quad (17)$$

$$P^d(t+\tau) = \frac{1}{Eff^d}\left(1 + \frac{1}{COP^d(t+\tau)}\right)\sum_{j\in J^d}\left(\left(P_j^p + \frac{P_j^0}{\phi^j}\right)\times \left(\hat{C}^{d,j}(t+\tau) + \sum_{i\in I(\tau)} pr_i(t+\tau)\hat{x}_i^{d,j}(t+\tau)\right)\right) \quad (18)$$

This problem can be solved by using the force-directed VM placement algorithm for the offline problem. Note that the number of VMs in this problem is limited to $|I(\tau)|$, which results in shorter execution time for this solution. It can be shown that, even starting from unsatisfactory background resource utilization, the online solution converges to a good solution after a number of iterations of the solution because the accuracy of the background workload will be improved by applying the online solution.

VII. Simulation Results

To show the effectiveness of the proposed algorithms for the GLB problem, a simulation framework is implemented.

In this simulation framework, we considered a US-based cloud system that has five datacenters in California, Texas, Michigan, New York, and Florida. The communication rate between these datacenters is assumed to be 1Gbps. Size of these datacenters rages from 4,000 to 1,600 servers belonging to four different server types, selected from HP server types. Duration of epoch is set to one hour. The average utilization of servers is assumed to be 70%. Peak power capacity of each datacenter is set to 80% of the peak power consumption of the deployed servers and cooling system. Based on the weather patterns, each datacenter has a combination of solar and wind power plant with

power generation capacity of up to 20% of its peak power consumption. The renewable power generation changes during the day based on type of the power plant. Energy price of each datacenter is assumed to follow the pattern shown in Figure 2 with appropriate time shift.

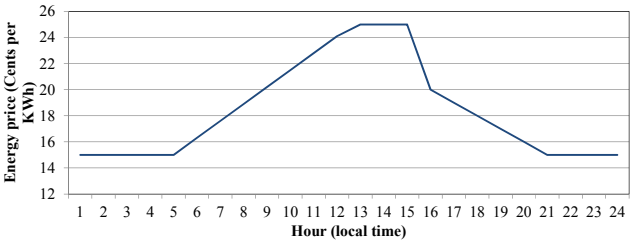

Figure 2 – Energy price offered by utility companies

To determine the relation between the COP and average power consumption in a datacenter, we applied the genetic-algorithm-based power provisioning policy presented in reference [18] to find the maximum COP for different range of power consumption in a two-row rack setting (250 blade servers with 110KW peak power) using hot-aisle/cold-aisle cooling arrangement. The results are reported in Figure 3.

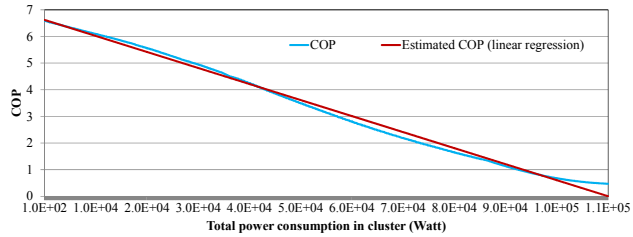

Figure 3 – Dependence of COP on average power consumption

It can be seen that the COP can be modeled as a linear function of the average power consumption with acceptable error. To approximate the COP function for the whole datacenter, the power coefficient in this linear estimation is multiplied by $p^{d,max}/110\text{KW}$. This assumption is based on having multiple server rooms with capacity of 110KW each.

Synthetic workloads are generated to be used in the GLB problem. Based on population distribution in US, applications are created in different time horizons and geographical locations. Application workload is changed according to the local time of its origination point. The application lifetime is set randomly based on uniform distribution between one and 16 hours. The SLA parameters and costs for these applications are set based on the Amazon EC^2 pricing scheme [26]. We used the SLA model presented in reference [16] to determine the SLA cost based on the amount of resources allocated to each VM. The minimum resource requirement for each VM is determined considering a target response time, a tolerable response time violation rate, behavior of the VM on the target server type, the round-trip time between VM location and target datacenter location, and the time required to transmit the typical packet in the incoming requests and outgoing responses. The penalty for an under-serviced request is set to be equal to the service price for one hour divided by the maximum number of requests that can violate the response time in each charge cycle. The migration cost is considered to be linearly related to the migration latency. The linear coefficient is set to be equal to the service price for one charge cycle divided by the worst possible migration latency (New York to California.)

The baseline in our simulation is a case without geographical load balancing. For this scheme, each client is assigned to its nearest datacenter that has sufficient available resources. This scheme results in low VM migration cost if there are no resource contentions in the datacenters.

To show the effectiveness of the proposed offline algorithm, we created workload for more than 100,000 clients across the US for a full day and determined the GLB solution by using proposed algorithm and baseline solution. The workload intensity, which is obtained by summing the minimum resource requirement for the active VMs, is reported in Figure 4.

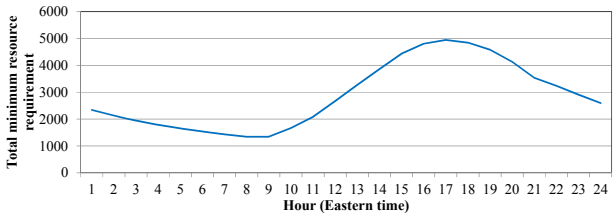

Figure 4 – The intensity of the workload as a function of time of the day captured by the total minimum resource requirement for active VMs

The operational cost of the cloud system with the FDLB algorithm, the baseline algorithm, and FDLB-1 (a simplified version of FDLB) are presented in Figure 5. Note that FDLB-1 constructively (i.e., epoch-by-epoch) determines the VM placement solution in order to reduce the run-time of the original solution.

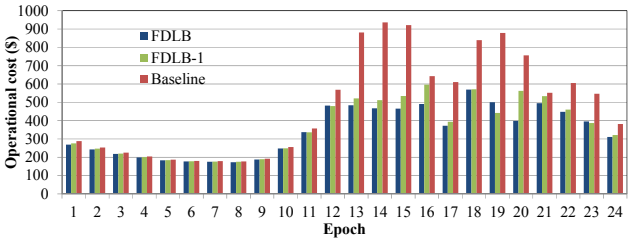

Figure 5 – Operational cost of the cloud in different epochs using different scheduling algorithms

As can be seen, in the beginning of the day, performance of the baseline method is similar to that of FDLB algorithm but in peak workload hours, the total operational cost using the baseline algorithm increases significantly. The total operational cost of the cloud system for one day by using the FDLB algorithm is 40% less than that of the baseline algorithm and 5% better than that of the FDLB-1. The run-time of FDLB, FDLB-1 and baseline on a 2.66GHz quad-core HP server are 466, 69 and 7 seconds, respectively. Share of different elements of the operational cost using FDLB algorithm is shown in Figure 6. As can be seen, FDLB solution avoids VM migration in light workload but VM migration is used under heavy workload situations to reduce the PUE, increase the share of renewable energy, and decrease the energy cost.

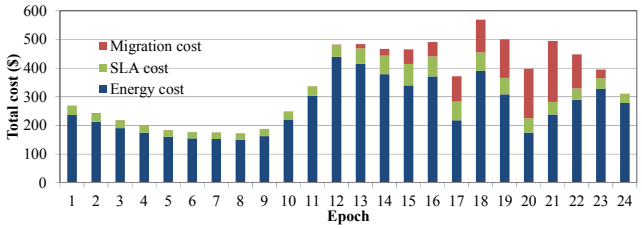

Figure 6 – Share of energy, SLA and migration cost in operational cost in different epochs

To show the effectiveness of the proposed online solution, we created a four-day scenario. To be able to apply the prediction about the background workload from first day of the online algorithm to the other days, we considered similar situations for all four days. The predicted parameters (discussed in section VI) are deviated from real values by up to 10% to model the misprediction phenomenon. The number of created VMs in each day is at least 100K. Normalized total operational costs of each day using the online and offline FDLB algorithms and the baseline method are reported in Table I. As can be seen, the online version of the algorithm works 8% worse than the complete and perfect information scenario in the offline version but it is 7% better than only considering the current epoch (FDLB-1) and 27% more effective than not considering the load balancing opportunity. Moreover the efficacy of the online algorithm improves by updating the background workload after the first day. Run-time of the online algorithm (after background workload preparation) ranges from 10 to 80 seconds for each epoch on a 2.66GHz quad-core HP server.

TABLE I. NORMALIZED TOTAL OPERATIONAL COST OF THE CLOUD DURING FOUR DAYS USING DIFFERENT LOAD BALANCING ALGORITHMS

Day	Normalized total OPEX for full day			
	Online FDLB	Offline FDLB	Online FDLB-1	Baseline
First day	1	0.91	1.02	1.24
Second day	1	0.92	1.08	1.29
Third day	1	0.92	1.08	1.29
Fourth day	1	0.94	1.09	1.31
Overall	1	0.92	1.07	1.27

It is worth noting that load balancing for online service applications is more effective when there are some resource contentions, different energy prices, or varying renewable power availabilities in datacenters' site. This fact is noticeable in Figure 5. The difference between FDLB and baseline results increases by having heavier workload in the cloud system. In contrast, decreasing the number of clients to half (50K) reduces the benefit of performing load balancing in the mentioned settings to 8% and 6% for the offline and online algorithms.

VIII. CONCLUSION

This work focused on the load balancing problem for online service applications considering a distributed cloud system comprised of geographically dispersed, heterogeneous datacenters. The problem formulation and a novel solution were presented and simulation results demonstrated the effectiveness of the proposed algorithms. The effectiveness of GLB was shown to be greater for high workloads and different electrical energy prices in datacenters' site. A possible future work is to combine the GLB problem for online applications with offline computation tasks scheduling problem to increase the benefit of the load balancing. Another possible future work is to consider GLB problem with multi-tier applications, which create multiple dependent VMs.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinsk, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "A view of cloud computing," *Commun ACM,* vol. 53, no. 4, pp. 50-58, 2010.

[2] R. Buyya, "Market-oriented cloud computing: Vision, hype, and reality of delivering computing as the 5th utility," in *9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID*, 2009.

[3] A. Ipakchi and F. Albuyeh, "Grid of the future," *IEEE Power and Energy Magazine,* vol. 7, no. 2, pp. 52-62, 2009.

[4] "http://www.google.com/green/energy/," [Online].

[5] R. Miller, "Facebook installs solar panels at new data center," DatacenterKnowledge, 16 April 2011. [Online].

[6] M. A. Adnan, R. Sugihara and R. Gupta, "Energy Efficient Geographical Load Balancing via Dynamic Deferral of Workload," in *proceeding of 5th IEEE conference on cloud computing (CLOUD 2012)*, Honolulu, HI, 2012.

[7] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," *ACM SIGMETRICS Performance Evaluation Review,* vol. 40, no. 1, 2012.

[8] M. Lin, Z. Liu, A. Wierman and L. L. Andrew, "Online algorithms for geographical load balancing," in *Proc. Int. Green Computing Conf.*, San Jose, CA, 2012.

[9] Z. Liu, M. Lin, A. Wierman, S. H. Low and L. L. H. Andrew, "Greening geographical load balancing," in *Proc. ACM SIGMETRICS*, San Jose, CA, 2011.

[10] K. Le, R. Bianchini, M. Martonosi and T. D. Nguyen, "Cost and energy-aware load distribution across data centers," in *HotPower'09*, Big Sky, MT, 2009.

[11] P. Paulin and J. Knight, "Force-directed scheduling for the behavioral synthesis of ASICs," *IEEE TRansaction on Computer-Aided Design of Integrated Circuits and Systems,* 1989.

[12] A. Verrna, P. Ahuja and A. Neogi, "pMapper: Power and migration cost aware application placement in virtualized systems," in *ACM/IFIP/USENIX 9th International Middleware Conference*, 2008.

[13] D. Ardagna, B. Panicucci, M. Trubian and L. Zhang, "Energy-Aware Autonomic Resource Allocation in Multi-Tier Virtualized Environments," *IEEE Transactions on Services Computing,* vol. 99, 2010.

[14] H. Goudarzi and M. Pedram, "Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems," in *proceeding of 4th IEEE conference on cloud computing (CLOUD 2011)*, 2011.

[15] H. Goudarzi and M. Pedram, "Maximizing profit in the cloud computing system via resource allocation," in *proc. of international workhop on Datacenter Performance*, 2011.

[16] H. Goudarzi, M. Ghasemazar and M. Pedram, "SLA-based Optimization of Power and Migration Cost in Cloud Computing," in *12th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, 2012.

[17] H. Goudarzi and M. Pedram, "Energy-efficient Virtual Machine Replication and Placement in a Cloud Computing System," in *IEEE international conference on cloud computing (CLOUD 2012)*, Honolul, 2012.

[18] Q. Tang, S. Gupta and G. Varsamopoulos, "Thermal-Aware Task Scheduling for Datacenters through Minimizing Heat Recirculation," in *Proc. IEEE Cluster*, 2007.

[19] L. Rao, X. Liu, L. Xie and W. Liu, "Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity market environment," in *IEEE INFOCOM*, 2010.

[20] R. Stanojevic and R. Shorten, "Distributed dynamic speed scaling," in *IEEE INFOCOM*, 2010.

[21] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," in *Proceedings of International Conference on Autonomic Computing (ICAC '08)*, 2008.

[22] X. Wang and M. Chen, "Cluster-level feedback power control for performance optimization," in *IEEE HPCA*, 2008.

[23] Z. Liu, M. Lin, A. Wierman, S. H. Low and L. L. H. Andrew, "Geographical load balancing with renewables," in *Proc. ACM GreenMetrics*, 2011.

[24] L. A. Barroso and U.Hölzle, "The Case for Energy-Proportional Computing," *IEEE Computer,* vol. 40, 2007.

[25] H. Goudarzi, S. Hatami and M. Pedram, "Demand-side load scheduling incentivized by dynamic energy prices," in *IEEE International Conference on Smart Grid Communications*, 2011.

[26] "http://aws.amazon.com/ec2/#pricing," [Online].