Green and Fair Workload Distribution in Geographically Distributed Datacenters

---

Rawan Y. Alawnah, Imtiaz Ahmad and Ebrahim A. Alrashed

Computer Engineering Department, College of Computing Sciences and Engineering, Kuwait University, Kuwait, imtiaz.ahmad@ku.edu.kw

Received: April 1, 2014; Accepted: April 27, 2014

Publication: May 25, 2014

Abstract

To accommodate user applications and requests, service providers host large computer systems over an infrastructure of distributed datacenters in different geographical locations. Consequently, energy and power consumption increases, increasing electrical and cooling costs. Hence more attention is paid to green computing and exploiting renewable energy sources, along with decreasing electrical costs so that these datacenters are operated efficiently and effectively, with less impact on the environment. In this paper, a socially responsible scheduling scheme that distributes load from several regions to geographically distributed datacenters will be extended. Optimal electricity, social and latency costs will be sought within service level agreements, ensuring fairness between different regions and highlighting its importance. Adopting standard gradient methods and Lagrange multipliers, an optimization algorithm called Green-Fair is proposed that provides optimal selection of datacenters. Experimental results show the effectiveness of the Green-Fair algorithm in utilizing green energy sources, reducing the number of SLA violations significantly, reducing latency costs and preserving fairness requirements by the SLAs while having little effect on the service providers' overall operational costs.

Keywords: Datacenter, green computing, cloud computing, SLA, fairness, optimization, Lagrange.

Journal of Green Engineering, Vol. 4, 69–98.

doi: 10.13052/jge1904-4720.415

© 2014 River Publishers. All rights reserved.

1 Introduction

Network based cloud computing is rapidly expanding as an alternative to conventional office-based computing [1], to support requests of clients who do not have means to set up and operate their own computing facilities [2]. A Cloud is a distributed system consisting of dynamically provisioned virtualized computers [3], delivering computing services and applications [4], such as shared resources, software, and information, through the internet to end users and end requests based on service-level agreements established through negotiations between consumer and service provider. As the cloud's network and computing resources grow, so does the energy and power consumption. It has been estimated that the power consumption in cloud datacenters increased 400% over the past decade [5]. A datacenter's monthly budget is mainly consumed by the servers' operational costs (53%) and power consumption (42%) [6 – 7]. The carbon – related global warming concern, along with large companies caring for their electrical budget, cost efficient and green resource allocation strategies among distributed datacenters are becoming important, resulting in the emergence of Green computing as a research area. “Green computing refers to environmentally sustainable computing which studies and practices virtually all computing efficiently and effectively with little or no impact on the environment” [8]. Datacenter owners and cloud providers are working on using power from renewable energy sources by buying electricity directly from wind and solar farms near their datacenters [2].

In order to support the vast customer demands, service providers host large computer systems over a number of distributed datacenters in the cloud, in different geographical locations, which each may use different electricity providers (some of which may be using renewable energy resources such as solar and wind energy). Throughout these different locations, the electricity costs may vary, and the locations of datacenters with lower electricity costs are favored. Most the research that has been done on energy planning for distributed datacenters are considering the electrical costs alone [9–12], renewable energy costs alone [13–15], and a few research combines both. The source of energy used by the datacenter should be taken into account, for more socially responsible [16] resource allocation. The social cost is determined by the extent of renewable energy usage [16] (the more renewable energy used, the less the social costs). A socially responsible scheduling scheme proposed by the authors in [16] that distributes load from several geographically distributed regions to geographically distributed datacenters will be extended. The scheme

distributes the load providing optimal electricity cost, latency cost and social cost.

The purpose of this research is to incorporate the importance of Service Level Agreements (SLAs) into the previous scheme. SLAs require that each datacenter performs efficiently, and is utilized effectively. An SLA includes the price of a service, and the QoS levels are adjusted by adjusting the price of the service [17]. These agreements provide a balance between the supply and demand of computing resources [3]. There are several aspects that may be agreed upon within SLAs, and one of these aspects (which is the focus of this research) is fairness amongst users of the cloud's services. As will be described in this paper, service requests that are closer in location to a datacenter are favored in the traditional load scheduling mechanisms, causing users in other locations to suffer lower quality of service standards. Quality of Service (QoS) standards and requirements vary with time, and so need to be dynamically updated [3]. Service agreements ensure fairness amongst consumers, providing service capacity constraints, and latency constraints. The Green-Fair algorithm proposed in this paper will consider along with the costs minimized in [16], the fairness of the load distribution. Optimal electricity, social and latency costs will be sought within fairness constraints agreed upon in the service level agreements.

The paper is organized as follows; in Section 2, related work is discussed. In Section 3, the problem formulation is described, and the proposed algorithm is depicted in Section 4. The simulation of the algorithm and experimental results are reported in Section 5. Finally, discussions and conclusions are provided in Section 6.

2 Related Work

There are many research efforts aimed to reduce the power consumption of geographically distributed datacenters [18–20], in order to reduce the overall electricity costs. Research efforts that exploit power management in the datacenter propose energy saving techniques on one of two levels; server level, where the power consumption of a single server is monitored, and at the datacenter level, where the entire datacenter and its servers are monitored for power consumption.

Considering the server layer, research approaches can be divided into three different operational layers, as described in [21]. Optimization techniques have been developed at the compiler layer, sleep mode (setting the processor to idle state when there are no requests) and dynamic voltage and frequency

scaling (DVFS) techniques (reducing the clock rate of the processor) have been developed and used at the operating system (OS) layer [22 – 23]. Virtual machines (VM) migration is used with server consolidation [24], and sleep mode techniques and power management schemes have been explored [21, 25]. A cloud monitoring infrastructure is proposed in [26] that timely collects power consumption from physical servers. Finally at the application layer, requests are served at fast speeds in order to complete the tasks quickly allowing the processor to finally be put into idle state. On the other hand, even though the overall energy consumption of datacenters has increased, for the past decade, the energy efficiency of servers (performance per watt) has typically doubled every two to four years [27].

As for the datacenter layer, most of the research efforts that aim to minimize power consumption involve virtualization and virtual machine migration. The authors in [18] aim to increase service providers' revenue by reducing their electricity costs by dynamically provisioning the servers. In [19], the authors provide both a server level power control and datacenter level load balancing scheme to reduce the electricity bill of the service provider. A RED-BL (Relocate Energy Demand to Better Locations) exploit the geo diversity in electricity priced markets [20] to provide optimal workload mapping. Many optimization problems have been formulated and optimization algorithms developed in the aim to comprise efficient task scheduling techniques [28–34], mostly considering the server layer, between servers, and not much attention has been paid to scheduling techniques at the datacenter layer, between datacenters.

Energy efficiency demands reduced usage of energy from fossil fuels [35]. The usage of renewable energy is encouraged, and several workload distribution algorithms are proposed [35–38] to utilize green energy. In [35], the feasibility of powering datacenter infrastructure using renewable energy is considered. A green-aware data selection algorithm is proposed in [37], which dynamically schedules workload across datacenters with respect to the dynamic variation of renewable energy sources.

In [36], the authors provide a clear flow diagram description of an algorithm that aims to reduce power consumption and carbon emission, with conformation to SLA constraints. A general optimization framework for datacenter selection is proposed in [39], and is mainly based on the concept of Nash bargaining concept in game theory. The SLA constraints do not allow a node to send all incoming requests to the most economically efficient datacenter, but rather to the datacenter with good performance. This provides a balance between minimizing costs for the service provider and

maximizing performance for the client or service requestor. An online efficient scheduling algorithm is proposed in [40] which aim to optimize energy cost and fairness amongst different organizations. The authors in [17] provide a Cloud computing framework that ensures SLAs are not violated, with location aware resource allocation to distributed datacenters. A power budgeting design based on the logical level and using distribution trees is proposed in [41]. The trees help differentiate and analyze the effect of workload types and Service Level Agreements (SLAs) which include response time, in terms of power characteristics.

An optimization algorithm will be developed to provide extension to the work in [16]. The Green-Fair algorithm will aim to reduce electrical, social and latency costs, within SLA fairness constraints, maintaining fairness amongst consumers. Previous works consider minimizing electrical costs, and others consider minimizing the social costs, reducing the carbon footprint. We consider here both these costs, along with the latency costs which ensure that the quality of service isn't affected by choosing datacenters further away from regions. We also consider fairness constraints, to ensure that SLA agreements are not violated. We will show that combining all these objectives together, (minimizing electrical, social and latency costs and maximizing fairness) will have little effect on the overall costs. That is, seeking green energy and maintaining fairness will not affect the operational costs significantly. The Green-Fair algorithm will distribute load to several geographically distributed datacenters, providing balance between demands from several regions.

3 Problem Formulation as an Optimization Problem

Before constructing the optimization procedure and writing the algorithm, first the problem must be formulated; it is a joint optimization problem with the following requirements:

- Minimize the performance cost.
- Minimize the social cost (more renewable energy usage).
- Minimize the electricity cost.
- Maximize fairness.

The following assumptions are made:

The entire service load is known a priori, and found by prediction techniques.

The total capacity of datacenters can accommodate the entire service load from all the regions (every request will be serviced).

The network flow model is shown in Figure 1, and the notations are depicted in Table 1. Load from M regions is to be distributed to N datacenters, where each datacenter is using a different kind of energy supply, and each energy supply differs from the other by its green factor.

Three costs are considered for minimization:

1. **Electricity cost**, which is the product of the amount of electricity used, and the electricity unit price. The electricity cost for N datacenters is defined as [16]:

$$EC = \sum_{j=1}^N \alpha_j \cdot e_j \quad (1)$$

where α_j is the unit price of electricity at datacenter j , and e_j is the energy consumption of datacenter j . The energy consumed by each server is modeled as follows:

$$E^{server}(v) = E_{idle}^{server} + (E_{peak}^{server} - E_{idle}^{server}) \cdot v \quad (2)$$

where $v \in [0, 1]$ is the load factor, a ratio between the actual load and the peak load of a server. The energy consumption of datacenter j is modeled as follows:

Figure 1 Load Distribution Model, showing how load λ_i received from each region is distributed by a load scheduler to the various datacenters. Each datacenter consumes energy from a different Electricity Providers (EP).

Table 1 Notation of relevant parameters to the problem formulation

M	number of user regions
N	number of geographically distributed datacenters
α_j	electricity unit price of Electricity Provider j (EP_j)
ϵ_j	green factor of energy supplied by EP_j
C_j	service capacity of datacenter j
λ_i	service load generated from region i
$k_{ij}(\cdot)$	latency cost between region i and datacenter j
γ	service capacity of one server
S_j	number of active servers in the j^{th} datacenter

$$e_j = E_j^{DC}(\mu_j) = S_j \cdot E^{server}(\mu_j/S_j) \quad (3)$$

where μ_j is the actual load and is defined as:

$$\mu_j = \sum_{i=1}^M \lambda_{ij}, \mu_j = S_j \gamma \quad (4)$$

Where S_j is the number of active servers at datacenter j, and γ is the server capacity, and their product is therefore the total capacity available.

2 Performance cost, which is computed as the latency cost. The latency cost is assumed to be determined by a non-decreasing convex function $k_{ij}(\cdot)$ [16]. The latency cost is defined as:

$$LC = \sum_{i=1}^M \sum_{j=1}^N k_{ij}(\lambda_{ij}) \quad (5)$$

where k_{ij} is the average latency and λ_{ij} is the amount of service load generated from region i to datacenter j. The latency cost function is defined as

$$k(x) = \frac{d}{1000} + \frac{1}{C - x} \quad (6)$$

where d is the geographical distance between a datacenter and a region, x is the total service load scheduled to a datacenter and C is the maximum service load capacity. Thus the equation is re-written as $(\lambda_{ij}) = \frac{d}{1000} + \frac{1}{C_j - \lambda_{ij}}$. The latency cost function varies since the total service load scheduled to a datacenter varies with time.

3 Social cost, which is the cost of using non renewable energy sources. Thus if more renewable energy is used, the less the cost. The social cost is defined as [16]:

$$SC = \sum_{j=1}^N \epsilon_j e_j \quad (7)$$

Where ϵ_j is the green degree of energy usage, which is a given or assumed value.

Load scheduling must also consider fairness and delay constraints. When multiple organizations or users share the same resources, allowing jobs from fewer users to run first while deferring other jobs for lower electricity prices may result in reducing energy cost [40], but may sacrifice the response time of other users, affecting fairness in the system. The aim of this research is to ensure fairness in the optimum load scheduling described above. The SLAs should provide requirements about how much computing services can be provided by a specific datacenter to a specific region. This ensures that a datacenter isn't being over used while other datacenters stay idle. It also ensures that a region is not favored by a datacenter allocation for its location. The authors in [40] define a fairness function to mathematically characterize the fairness of resource allocation;

$$f(t) = - \sum_{q=1}^Q \left[\frac{r_q(t)}{R(t)} - \gamma_q \right]^2 \quad (8)$$

where $r_q(t)$ is the total amount of computing resources allocated to all jobs coming from account q during time t , $R(t)$ is the total available computing resources during time t , and $\gamma_q \geq 0$ is the weighting parameter indicating the desired amount of resource allocation to account q . The fairness score is maximized when $r_q(t) = \gamma_q R(t)$ for all q ($q = 1, 2, \dots, Q$), where Q is the total number of accounts. Equation 8 can be rewritten using the parameters described in Table 1:

$$f(\lambda) = - \sum_{i=1}^M \left[\frac{\lambda_{ij}}{C_j} - \gamma_{ij} \right]^2 \text{ for all data centers, or}$$

$$f(\lambda) = - \sum_{j=1}^N \sum_{i=1}^M \left[\frac{\lambda_{ij}}{C_j} - \gamma_{ij} \right]^2 \quad (9)$$

where M is the number of regions, λ_{ij} is the total service load generated from region i to datacenter j , C_j is the total capacity of datacenter j , and γ_{ij} is a weighting parameter for each region stating the desired amount of service load allocation to region i from datacenter j , defined by SLAs. The fairness is maximized when $\lambda_{ij} = \gamma_{ij} C_j \forall i, j$.

After defining the costs and the fairness function, the problem can now be formulated to find the final function to be minimized.

It is possible to write an optimization problem in the generic form [42]:

$$\begin{aligned} & \text{minimize}_{x \in \mathbb{R}^n} f_i(x), \quad (i = 1, 2, \dots, M), \\ & \text{subject to } \epsilon_j(x) = 0, \quad (j = 1, 2, \dots, J), \\ & \psi_k(x) \leq 0, \quad (k = 1, 2, \dots, K), \end{aligned}$$

where $f_i(x)$, $\phi_i(x)$ and $\psi_i(x)$ are functions of the design vector $x = (x_1, x_2, \dots, x_n)^T$, where the components x_i of x are called design or decision variables. The functions $f_i(x)$ are called the objective functions. In literature, the objective function is usually called the energy or cost function. There are several ways to classify an optimization problem:

By the number of objectives; if the problem consists of one or more objectives, it is a multi-objective optimization problem. For example, a load scheduling optimization problem may have the objectives to minimize electricity cost, and maximize performance.

By the number of constraints; if the problem is subject to one or more constraints, it is a constrained optimization problem, otherwise it is unconstrained.

By the form of the objective functions and constraints; if the constraints are linear, it is a linearly constrained problem. If both constraints and objective functions are linear, then it becomes a linear programming problem.

For different optimization problems, we often have to use different optimization techniques [42], since some algorithms are more suitable than others for certain problems. Optimization algorithms can be classified into two types; deterministic and stochastic, the latter being more random. Most classic algorithms are deterministic, and those that use gradient information are called gradient-based algorithms, which will be used in the Green-Fair algorithm proposed. Multi-objective optimization problems do not necessarily have an optimum solution that minimizes all the objective functions at the same time. Therefore, there is always a tradeoff when solving multi-objective minimization problem, and objectives can be classified based on priority. Many solution algorithms intend to combine all the multi-objective functions into one scalar objective using the weighted sum [42] shown in Equation 10 below:

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \dots + \alpha_p f_p(x) \quad (10)$$

where each objective function is multiplied by a coefficient or weight factor. The solution to the problem therefore depends strongly on the weighting coefficients.

We adopt the method used in [40] to formulate the optimization problem as a weighted function of Electricity, Social and Latency cost with Fairness, and call it GF:

$$GF = EC + LC + SC - \beta \cdot f(\lambda) \quad (11)$$

The problem is thus formulated as follows:

Problem GF: $\min EC + LC + SC - \beta \cdot f(\lambda) = C(\lambda) - \beta \cdot f(\lambda)$
under the following three constraints:

1. $\sum_{i=1}^M \lambda_{ij} \leq C_j, \forall_j$ (Service load going to datacenter j should be less than or equal to capacity of the datacenter)
2. $\sum_{j=1}^N \lambda_{ij} = \lambda_i, \forall_i$ (All service load generated from any region will be distributed to feasible datacenters)
3. $\lambda_{ij} \geq 0, \forall_{i,j}$ (The amount of service load from any region directed at a datacenter should be greater than or equal to 0)

Given that;

- The capacity of a datacenter j is C_j
- The service load for datacenter j is μ_j
- The minimal number of required active servers at a datacenter is μ_j/γ ,

The first term of GF is given as:

$$\begin{aligned} C(\lambda) &= EC + LC + SC \\ &= \sum_{j=1}^N \alpha_j \cdot e_j + \sum_{i=1}^M \sum_{j=1}^N k_{ij}(\lambda_{ij}) + \sum_{j=1}^N \epsilon_j e_j \end{aligned}$$

The second term of the equation is given as:

$$\beta f(\lambda) = \beta \cdot \sum_{i=1}^M \sum_{j=1}^N \left[\frac{\lambda_{ij}}{C_j} - \gamma_{ij} \right]^2$$

Since $e_j = E_j^{DC}(\mu_j)$ and $\mu_j = \sum_{i=1}^M \lambda_{ij}$, then the optimization problem can be re-written as a function (F) of load:

$$F(\lambda) = \sum_{j=1}^N E_j^{\text{DC}} \left(\sum_{i=1}^M \lambda_{ij} \right) \cdot (\alpha_j + \epsilon_j) + \sum_{i=1}^M \sum_{j=1}^N k_{ij}(\lambda_{ij}) \\ + \beta \cdot \sum_{i=1}^M \sum_{j=1}^N \left[\frac{\lambda_{ij}}{C_j} - \gamma_{ij} \right]^2 \quad (12)$$

As can be seen from Equation 12, the minimization problem is a constrained problem subject to several constraints, with the multi-objective of minimizing cost and maximizing fairness. The constraints are linear however the solution must be iterative, converging to an optimal solution. The problem is thus a multi-objective linearly constrained optimization problem. The proposed algorithmic solution (Green-Fair) is a deterministic gradient-based one.

Since the problem is a constrained optimization problem, with multiple constraints, the aim is to transform the problem into an unconstrained one and accordingly, the Lagrangian is found. The Lagrangian is defined as the sum of the original objective function and a linear combination of the constraints [43]. If we want to minimize a function $\min_{\mathbf{x} \in \mathcal{R}^n} f(\mathbf{x})$, $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathcal{R}^n$, subject to the constraint $g(\mathbf{x}) = 0$ [42], then the objective function and the constraint can be combined to form a new function called the Lagrangian;

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x}) \quad (13)$$

where f is the function to be minimized, g is the combination of constraints and λ is the Lagrange multiplier, an unknown scalar to be determined. This method can be extended to nonlinear optimization with multiple equality constraints.

The factor β in Equation 11 is equivalent to the Lagrangian multiplier if we formulate the fairness as a constraint [44];

$$4. \lambda_{ij} = \gamma_{ij} C_j, \forall_{i,j}, \gamma_{ij} \geq 0$$

Thus the Lagrangian function for Problem GF is written as follows:

$$L(\lambda, \theta, \eta, \varphi) = C(\lambda) + \sum_{j=1}^N \theta_j \cdot \left(\sum_{i=1}^M \lambda_{ij} - C_j \right) \\ + \sum_{i=1}^M \eta_i \cdot (\lambda_i - \sum_{j=1}^N \lambda_{ij}) + \sum_{i=1}^M \sum_{j=1}^N \varphi_{ij} \cdot (\lambda_{ij} - \gamma_{ij} C_j) \quad (14)$$

where $\theta_j \geq 0, \lambda_{ij} \geq 0, \varphi_{ij} \geq 0, \theta = \{\theta_j, j \in [1, N]\}, \eta = \{\eta_i, i \in [1, M]\}, \varphi = \{\varphi_{ij}, i \in [1, M], j \in [1, N]\},$ and $k_{ij} = \frac{d}{1000} + \frac{1}{C_j - \lambda_{ij}}$

The first term of Equation 14 is the main cost function that is to be minimized. The second term is the first constraint multiplied by a Lagrangian multiplier, and the third term is the second constraint multiplied by another Lagrangian multiplier. Finally the last term is the third fairness constraint, multiplied by a third Lagrangian multiplier. The costs of violating the constraints are priced by the Lagrangian multipliers.

The problem GF is solved using dual composition. The primal problem (previous) is a minimization problem, hence the dual problem will be a maximization problem. The Lagrangian is re-arranged and re-written as follows:

Let $\left(E_{idle}^{server} \cdot \frac{1}{\gamma} + (E_{peak}^{server} - E_{idle}^{server})\right) \cdot (\alpha_j + \epsilon_j) = g_i$, and since $\mu_j = \sum_{i=1}^M \lambda_{ij}$ then;

$$\begin{aligned} L(\lambda, \theta, \eta, \varphi) &= \sum_{j=1}^N g_i + \sum_{i=1}^M \sum_{j=1}^N k_{ij}(\lambda_{ij}) + \\ &\quad \sum_{j=1}^N \theta_j \cdot \left(\sum_{i=1}^M \lambda_{ij} - C_j \right) + \sum_{i=1}^M \eta_i \cdot (\lambda_i - \sum_{j=1}^N \lambda_{ij}) \\ &\quad + \sum_{i=1}^M \left(\sum_{j=1}^N \varphi_{ij} \cdot (\lambda_{ij} - \gamma_{ij} C_j) \right) \\ &= \sum_{i=1}^M \left\{ \sum_{j=1}^N (g_i + \theta_j - \eta_i) \cdot \lambda_{ij} + k_{ij}(\lambda_{ij}) + \varphi_{ij} \cdot (\lambda_{ij} - \gamma_{ij} C_j) \right\} - \\ &\quad \sum_{j=1}^N \theta_j \cdot C_j + \sum_{i=1}^M \eta_i \cdot \lambda_i \end{aligned}$$

The multipliers in this case are vector not scalar, so the problem is solved as a dual problem. Duality arises in nonlinear and linear optimization models in a wide variety of settings [45], and it is often used in the convergence analysis of algorithms. The main idea in duality is that the primal problem has an associated dual problem where the variables are the Lagrange multipliers of the primal constraints [46]. The number of dual variables is the same as the number of primal constraints, and the number of dual constraints is the same as the number of primal variables. The minimization

problem of the original problem becomes a maximization problem in the dual problem. The primal and dual variables both satisfy the non-negativity condition [43]. The dual problem is a concave maximization problem [45]. The optimal primal design variables are obtained iteratively until convergence is achieved.

Since the variable λ_{ij} or load is not a vector but a matrix, each element represents the loads distributed from a region i to a datacenter j . The rows therefore represent all the loads distributed from region i , and each column represents all the loads distributed to datacenter j . The objective can be reached by finding the optimum load distribution from each region (therefore each row or i^{th} row), updating the regions and datacenters with this information, and moving on to the next row. The iteration is repeated until an optimal solution for all the rows is found (i.e. until convergence is achieved). The i^{th} Lagrangian multiplier (minimization problem for each region) is defined as:

$$L_i(\lambda_i, \theta, \eta_i, \varphi_i) = \sum_{j=1}^N (g_i + \theta_j - \eta_i) \cdot \lambda_{ij} + k_{ij}(\lambda_{ij}) + \varphi_{ij} \cdot (\lambda_{ij} - \gamma_{ij} C_j) \quad (15)$$

where $\lambda_i = \{\lambda_{ij}, j \in [1, N]\}$ and $k_{ij} = \frac{d}{1000} + \frac{1}{C_j - \lambda_{ij}}$.

For given θ and η_i, φ_i , λ_i^* is defined as:

$$\lambda_i^* = \arg \min_{\lambda_i \geq 0} [L_i(\lambda_i, \theta, \eta_i, \varphi_i)] , \forall i$$

The dual (maximization) problem for GF is then given by:

$$h(\theta, \eta, \varphi) = \sum_{i=1}^M h(\lambda_i, \theta, \eta_i, \varphi_i) - \sum_{j=1}^N \theta_j \cdot C_j + \sum_{i=1}^M \eta_i \cdot \lambda_i \quad (16)$$

s.t $\theta \geq 0$ and $h(\lambda_i, \theta, \eta_i, \varphi_i) = L(\lambda_i^*, \theta, \eta_i, \varphi_i)$

The following three equations will be used in the Green-Fair algorithm, and will be updated within each iteration until convergence is achieved. At each iteration, the minimization procedure is performed, and certain data is recorded for the next iteration, or for termination.

$$\theta_j(t+1) = \left[\theta_j(t) - \rho \cdot \left(\sum_{i=1}^M \lambda_{ij} - C_j \right) \right]^+ \quad (17)$$

$$\eta_i(t+1) = \eta_i(t) - \omega \cdot \left(\lambda_i - \sum_{j=1}^N \lambda_{ij} \right) \quad (18)$$

$$\varphi_j(t+1) = \varphi_j(t) - \sigma \cdot (\lambda_{ij} - \gamma_{ij} C_j) \quad (19)$$

Where ρ, ω, σ are sufficiently small step sizes. The above three equations will be updated in the offline optimization algorithm, after every step. Equations 17 and 18, proposed in [16] represent the Capacity and Consistency constraints respectively. Equation 17 is updated by each datacenter, according to its capacity, and the service load it is receiving. Equation 18 is updated by each region, according to the region's total service load, and service load directed to each datacenter. The last equation, Equation 19 we proposed represents the Fairness constraint. It is also updated by each datacenter j , according to the service load it is receiving from a region, and the weighting parameter for that region, provided by the SLAs.

4 Green-Fair Algorithm

Indirect methods for solving constrained optimization problems are used to take advantage of codes that solve unconstrained optimization problems, and are referred to as Sequential Unconstrained Minimization Techniques (SUMT). The idea behind the approach is to repeatedly call the unconstrained optimization algorithm using the solution of the previous iteration [43]. The unconstrained algorithm executes several iterations itself. Prior to the processing of the algorithm, the constrained problem must be transformed to an unconstrained problem. The Augmented Lagrange Multiplier algorithm is one of the most robust solutions to an unconstrained optimization problem, as it also provides information about the Lagrangian multipliers.

The Green-Fair algorithm proposed follows a standard gradient method for optimization. The algorithm is offline, and is compared to the offline algorithm proposed in [16]. It is iterative, where the Lagrange multipliers are updated and the minimization is performed at each iteration. The algorithm is run at each time t , and the lambda (service load) from each region at time t is determined using a standard Gaussian distribution. The algorithm is shown in Figure 2. The algorithm is distributed, since the datacenters and regions are involved in updating the Lagrange multipliers. The minimization procedure is then performed by a centralized load scheduler, which uses the values obtained from the datacenters and regions to compute an optimum lambda and optimal function cost. The service load from each region is dynamic, and therefore will change with each time epoch.

Algorithm 1 – Green-Fair

Input: $C_j, \forall j$; $\lambda_i, \forall i$; $E_{peak}^{server}, E_{idle}^{server}$; $\alpha_j, \epsilon_j, \forall j; \gamma$

Output: Optimal load $\lambda_{i,j}, \forall i,j$, Optimal total cost $f(\lambda)$;

1. Initialization: set $\theta_j, \eta_i, \varphi_j$ all to 0.
2. Compute predicted service load for each region using Gaussian distribution.
3. Perform minimization using initial values of multipliers and record current results.
4. Each datacenter j updates θ_j and broadcasts result to each region.
5. Each region updates η_i and broadcasts result to each datacenter.
6. Load scheduler updates φ_j based on values obtained from SLAs.
7. **if** (maximum number of iterations reached)
  - a. Stop.
- else**
  - b. Set $t = t+1$ and repeat minimization procedure (step 3) until the convergence criteria is satisfied. (Stop if $f(\lambda)_t \approx f(\lambda)_{t-1}$)
8. **return:** $\lambda_{i,j}$ matrix and $f(\lambda)$

Figure 2 Green-Fair algorithm

At each time epoch, to the algorithm will run to find the optimal load distribution and terminate once the convergence criteria is satisfied. Once the algorithm starts to converge too slowly, or once there is insignificant difference between the values of the cost in the current and the previous iteration, the algorithm will terminate and return the value of the minimal cost obtained. If the algorithm does not satisfy the convergence criteria but has reached a maximum number of iterations previously defined, then it will also terminate and return the value of the minimal cost obtained.

5 Simulation Environment and Experimental Results

Regional electricity prices were obtained from the US electricity market [47] to be used in the simulation environment. A sample of the data and prices obtained is shown in Table 2. In the experiment, 15 regions and 8 datacenters were included. The SLA requirements were obtained with respect to the geographical distances between the regions and datacenters [47]. The SLAs as mentioned in this paper may dynamically change, however they are fixed for each region for the purpose of this experiment, and it is assumed that the total capacity of all the datacenters is enough to service the total load from all the regions. As mentioned above, the lambda (service load) from each region at time t is determined using a standard Gaussian distribution. The initial load for each region is previously assumed, and they are similar to the values used

Table 2 U.S. Regional Electricity Prices (Cents per Kilowatt-Hour)

Region	Regional Electricity Price		
	2012	2013	2014
New England	14.07	14.18	14.28
Middle Atlantic	12.77	13.13	13.39
E. N. Central	9.27	9.37	9.46
W. N. Central	8.50	8.59	8.70
S. Atlantic	9.71	9.79	10.00
E. S. Central	8.51	8.73	9.01
W. S. Central	8.14	8.58	8.80
Mountain	8.79	9.06	9.30
Pacific	11.51	11.51	11.77
U.S. Average	9.85	10.03	10.23

in [16]. The mean and variance for the distribution are then determined and indicated in Table 3, other constants and variables are also listed with their corresponding values used in the experiment. The ranges of values for the social cost and distance, along with the Energy values for each server are all obtained from [16].

The Green-Fair offline algorithm proposed is compared with the offline Socially Responsible algorithm proposed in [16]. We also compare our algorithm with the Cheap-First strategy, which selects the datacenter with the least electricity cost to serve a region, therefore only considering electricity costs, and neglecting social and latency costs. In the Figures referenced Figure 3 – Figure 15, the Socially Responsible algorithm [16] is referred to as the “Green” algorithm, whereas the proposed algorithm is referred to as “Green-Fair”, and finally the Cheap –First strategy is referred to as Cheap-First.

As can be seen from the cost comparison Figures 3 to 6, the algorithms behave differently. The Cheap-First strategy imposes a large Social cost, higher than that of the Green and Green-Fair algorithms. The Green-Fair algorithm shows the least Social cost, as can be seen in Figure 3. Since the Cheap-First strategy always favors the datacenter with the least electrical cost, this strategy results in a lower electricity cost than the Green and Green-Fair algorithms, as can be seen in Figure 4. Figure 5 shows the difference in Latency cost between the algorithms. The Green-Fair shows the most favorable results here, since the SLAs will balance the load between regions and datacenters that are closer to each other. Figure 6 shows the Total cost (Electrical, Latency and Social combined) for the three algorithms, and it can be seen that the values are almost similar. These results show that the consideration of Social and

Table 3 Simulation Environment setting (values for variables and constants used in the experiment)

Constant/ Variable	Value
Mean & Variance (region 1)	~10000–15000 , 200–500
Mean & Variance (region 2)	~4000–7000 , 100–200
Mean & Variance (region 3)	~6000–10000 , 300–500
E_{idle}^{server}	60 (according to [16])
E_{peak}^{server}	100 (according to [16])
Capacity of any datacenter	20000
$k_{ij}(x) = \frac{d}{1000} + \frac{1}{C-x}$	latency cost between region i and datacenter j [16]
γ (Server service capacity)	10
ϵ_j	Social cost (1–10)
d (distance between region and datacenter)	~ (1–100)

Figure 3 Comparison of Normalized Social cost for the three algorithms

Latency costs, along with maintaining SLA requirements, does not increase the Total cost significantly.

The workload distributions throughout the 8 datacenters are shown in Figures 7, 8 and 9. The Green algorithm provides distribution to all datacenters; however, a couple of datacenters are highly overloaded while others receive much less load. In Figure 8, the Green-Fair algorithm provides more even distribution between the datacenters, with only one datacenter loaded to its maximum capacity. The Cheap-First strategy shows the worst load distribution as shown in Figure 9, where datacenters 3 to 8 are loaded to their maximum capacity, and no load has been distributed to datacenters 1 and 2, leaving them idle.

Figure 4 Comparison of Normalized Electricity cost for the three algorithms

Figure 5 Comparison of Normalized Latency cost for the three algorithms.

Figure 10 shows the datacenter capacity usage of datacenter 4 by regions 1 to 5. Datacenter 4 has the lowest electricity price, and as can be seen from the Figure, the Cheap-First strategy has exceeded the SLA restrictions by a large amount leading to a high number of SLA violations. Figure 11 shows the datacenter capacity usage of datacenter 8 by regions 6 to 10. Both the Cheap-First strategy and the Green algorithm significantly exceed the SLA

Figure 6 Comparison of Normalized Total cost for the three algorithms

Figure 7 Workload distributions throughout the 8 datacenters for the Green algorithm.

Figure 8 Workload distributions throughout the 8 datacenters for the Green-Fair algorithm.

Figure 9 Workload distributions throughout the 8 datacenters for the Cheap-First algorithm

Figure 10 Capacity usage of datacenter 4 by regions 1 to 5.

restriction while the Green-Fair algorithm only shows a slight increase to the amount of load allowed from region 10 to datacenter 8.

Figure 12 shows how the load is distributed from regions 6 to 10, to datacenter 1. Datacenter 1 has the highest regional electricity cost and so as can be shown in the Figure the Cheap-First strategy does not distribute any load to that datacenter. For region 6, the Figure shows that Green algorithm exceeding the factor required by the SLAs by more than double the amount. In Figure 13, even though datacenter 3 has a lower regional electricity unit price than most the other datacenters, the Green and Green-Fair algorithm

Figure 11 Capacity usage of datacenter 8 by regions 6 to 10

Figure 12 Capacity usage of datacenter 1 by regions 6 to 10

do not violate the SLAs, while the Cheap-First strategy exceeds the SLA requirements by up to a factor of 5.

The SLA violations that exceed the factors proposed by the SLA for the three algorithms are shown in Figure 14. There is a significant difference between the few violations committed by the Green-Fair algorithm, and those committed by the Green algorithm and the Cheap-First strategy.

Each datacenter has its own source of energy, and each energy supply has a green factor, associated with the type of energy used. The lower

Figure 13 Capacity usage of datacenter 3 by regions 6 to 10

Figure 14 Number of SLA violations committed by the three algorithms

the factor, the greener or more socially responsible the energy consumed. Figure 15 shows that the Green-Fair algorithm, with an average green factor usage of almost 5.6, uses more socially responsible energy supplies than the other two algorithms. The Cheap-First strategy has the highest factor (6.5), and the Green algorithm falls in between the Green-Fair and Cheap-First with an average factor of 6.

Figure 15 Average green factor of energy consumed for the three algorithms

6 Conclusions

This research provides a Green-Fair scheme that provides the workload distribution from customers in distributed regions, to a network of distributed datacenters. The workload is distributed based on results obtained from an optimization procedure, which finds the optimal load distribution that has the minimal Electricity, Social and Latency costs, as the authors in [16]. The optimization is subjected to SLA fairness constraints, maintaining agreements about latency and capacity usage between regions and corresponding datacenters. Thus clients will receive the quality of service agreed upon, and capacity of datacenters utilized by each region will be constrained within the capacity requirements established in the SLAs, ensuring fairness between regions. Optimal electricity, social and latency costs were sought within fairness constraints.

The results as discussed in Section 5 show that the Green-Fair algorithm improve the algorithm proposed in [16], and the Cheap-First strategy on a few levels; the work is distributed to all datacenters, and no datacenters are overloaded while others are idle; the number of SLA violations decreased significantly, and the overall energy used by the Green-Fair algorithm is greener. The algorithm also shows a decrease in the Social and Latency costs, with only a slight difference in the total cost; that is, with conforming to SLAs and using renewable energy sources, there can be little effect on the total cost.

As for future work, other SLA agreements beside fairness could be studied and sustained, including response time and throughput. Furthermore, the selection of data centers may be aided by collecting more detailed information

such as the number of active servers, their capacity, and if any servers need to be turned on or off. This will allow both a datacenter and a server level analysis, so server consolidation techniques could be considered, enhancing the energy efficiency of datacenters.

References

- [1] The NIST Definition of Cloud Computing, National Institute of Science and Technology, <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>. Retrieved 24 July 2013
- [2] M. Pedram. Energy Efficient Datacenters. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp 1465–1485, 2012.
- [3] V. Cardellini, E. Casalicchio, K. Branco, J. Estrella, and F. Monaco. Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions. IGI Global, doi : 10.4018/978-1-60960-794-4, 2012.
- [4] R. Katz, G. Lee, M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, A. Konwinski, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the Clouds: A Berkeley View of Cloud Computing. *Communications of the ACM*, 53(4):50–58, 2010.
- [5] H. Qian, F. Li, and D. Medhi. On Energy – Aware Aggregation of Dynamic Temporal Demand in Cloud Computing. In *Proc. Of IEEE Fourth International Conference on Communication Systems and Networks, COMSNETS, Bangalore, India*, pp 1–6, 2012.
- [6] J. Baliga, R. W. A. Ayre, and K. Hinton. Green Cloud Computing: Balancing Energy in Processing, Storage and Transport. In *Proc. of the IEEE*, pp 149–167, 2010.
- [7] J. Hamilton. Cooperative Expendable Micro-Slice Servers (CEMS): Low Cost, Low Power Servers for Internet-Scale Services. In *Proc. of Conference on Innovative Data Systems Research (CIDR’09)*, 2009.
- [8] Q. Dan and L. Green. Computing Methodology for Next Generation Computing Scientists. In *Proc. IEEE 34th Annual Computer Software and Applications Conference (COMPSAC)*, pp 250–251, 2010.
- [9] Y. Zhang, and Y. Wang, and X. Wang. Electricity Bill Capping for Cloud-Scale Datacenters that Impact the Power Markets. In *Proc. 41st International Conference on Parallel Processing (ICPP)*, pp 440–449, 2012.

- [10] J. Li, Z. Li, K. Ren, and X. Liu. Towards Optimal Electric Demand Management for Internet Datacenters. *IEEE Transactions on Smart Grid*, pp 183–192, 2012.
- [11] A. Qureshi, R. Weber, H. Balakrishnan, J. Gutttag, and B. Maggs. Cutting the Electric Bill for Internet scale Systems. In *Proc. of the ACM SIGCOMM Conference on Data Communication*, 2009.
- [12] L. Rao, X. Liu, L. Xie, and W. Liu. Minimizing Electricity Cost: Optimization of Distributed Internet Datacenters in a Multi-Electricity-Market Environment. In *Proc. of IEEE INFOCOM*, pp 1–9, 2010.
- [13] C. Ren, D. Wang, B. Urgaonkar, and A. Sivasubramaniam. Carbon-Aware Energy Capacity Planning for Datacenters. In *Proc. IEEE 20th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp 391–400, 2012.
- [14] N. Deng, C. Stewart, D. Gmach, M. Arlitt, and J. Kelley. Adaptive Green Hosting. In *Proc. of the 9th International Conference on Autonomic Computing (ICAC)*, pp 135–144, 2012.
- [15] K. Le, R. Bianchini, T. D. Nguyen, O. Bilgir, M. Martonosi. Capping the Brown Energy Consumption of Internet Services at Low Cost. In *Proc. International Green Computing Conference*, pp 3–14, 2010.
- [16] J. He, X. Deng, D. Wu, Y. Wen, and D. Wu. Socially-Responsible Load Scheduling Algorithms for Sustainable Datacenters over Smart Grid. In *Proc. 3rd IEEE International Conference on Smart Grid Communications (IEEE SmartGridComm)*, Taiwan, 2012.
- [17] S. Son, G. Jung, and S. C. Jun. A SLA-based Cloud Computing Framework: Workload and Location Aware Resource Allocation to Distributed Datacenters in a Cloud. In *Proc. International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA2012)*, Las Vegas, USA, 2012.
- [18] M. Mazzucco, D. Dyachuk, and R. Deters. Maximizing Cloud Providers Revenues via Energy Aware Allocation Policies. In *Proc. IEEE 3rd International Conference on Cloud Computing (CLOUD)*, pp 131–138, 2010.
- [19] L. Rao, X. Liu, M. Ilic, and J. Liu. MEC-IDC: Joint Load Balancing and Power Control for Distributed Internet Datacenters. In *Proc. 1st ACM/IEEE International Conference on Cyber-Physical Systems*, pp 188–197, 2010.

- [20] M. Ilyas, S. Raza, C. C. Chen, Z. Uzmi, and C. N. Chuah. RED-BL: Energy Solution for Loading Datacenters. In Proc. of IEEE INFOCOM, pp 2866–2870, 2012.
- [21] I. Sarji, C. Ghali, A. Chehab, and A. Kayssi. CloudESE: Energy Efficiency Model for Cloud Computing Environments. In Proc. 2011 International Conference on Energy Aware Computing (ICEAC), pp 1–6, 2011.
- [22] E. L. Sueur and G. Heiser. Slow Down Or Sleep, That Is The Question. In Proc. USENIX Annual Technical Conference, pp 16–16, 2011.
- [23] E. L. Sueur and G. Heiser. Dynamic Voltage and Frequency Scaling: The Law of Diminishing Returns. In Proc. of the Workshop on Power Aware Computing and Systems (HotPower’10), Article No. 1–8, 2010.
- [24] M. A. Vouk. Cloud Computing – Issues, Research and Implementations. In Proc. 30th International Conference on Information Technology Interfaces, ITI 2008, pp 31–40, 2008.
- [25] V. K. M. Raj and R. Shriram. Study on Server Sleep State Transition to Reduce Power Consumption in a Virtualized Server Cluster Environment. In Proc. 4th International Conference on Communication Systems and Networks (COMSNETS), pp 1–6, 2012.
- [26] A. Corradi, M. Fanelli, and L. Foschini. Increasing Cloud Power Efficiency through Consolidation Techniques. In Proc. the First Workshop on Management of Cloud Systems, IEEE Symposium on Computers and Communications (ISCC), pp 129–134, 2011.
- [27] J. G. Koomey, C. Belady, M. Patterson, A. Santos, and K. D. Lange. Assessing Trends Over Time In Performance, Costs and Energy Use for Servers. Lawrence Berkeley National Laboratory, Stanford University, Microsoft Corporation, and Intel Corporation, Tech. Rep., 2009.
- [28] Z. Wang and Y. Zhang. Energy-Efficient Task Scheduling Algorithms with Human Intelligence Based Task Shuffling and Task Relocation. In Proc. IEEE/ACM International Conference on Green Computing and Communications (GreenCom), pp 38–43, 2011.
- [29] N. A. Mehdi, A. Mamat, A. Amer, and Z. T. Abdul-Mehdi. Minimum Completion Time for Power-Aware Scheduling in Cloud Computing. In Proc. IEEE Developments in E-Systems Engineering (DeSE), pp 484–489, 2011.
- [30] L. M. Zhang, K. Li, and Y. Q. Zhang. Green Task Scheduling Algorithms with Speeds Optimization on Heterogeneous Cloud Servers. In Proc.

- IEEE/ACM International Conference on Green Computing and Communications (GreenCom) & International Conference on Cyber, Physical and Social Computing (CPSCoM), pp 76–80, 2010.
- [31] N. A. Mehdi, A. Mamat, A. Amer, and Z. T. Abdul-Mehdi. Two-Phase Provisioning for HPC Tasks in Virtualized Datacenters. In Proc. International Conference on Emerging Trends in Computer and Electronics Engineering (ICETCEE), Dubai, 2012.
  - [32] N. Liu, Z. Dong, and R. Rojas-Cessa. Task and Server Assignment for Reduction of Energy Consumption in Datacenters. In Proc. IEEE 11th International Symposium on Network Computing and Applications, pp 171–174, 2012.
  - [33] P. Tiwari, K. Jintamuttha, K. Manakul, and T. Achalakul. Process Placement Scheme Optimization for Energy-Efficient Datacenter. In Proc. 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp 1–4, 2012.
  - [34] L. Xu, Z. Zeng, and X. Ye. Multi-objective Optimization Based Virtual Resource Allocation Strategy for Cloud Computing. In Proc. IEEE/ACIS 11th International Conference on Computer and Information Science, pp 56–61, 2012.
  - [35] Z. Liu, M. Lin, and L. L. H. Andrew. Geographical Load Balancing with Renewables. ACM SIGMETRICS Performance Evaluation Review, pp 62–66, 2011.
  - [36] D. M. Quan, A. Somov, and C. Dupont. Energy Usage and Carbon Emission Optimization Mechanism for Federated Datacenters. Book Title: Energy Efficient Datacenters, pp 129–140, 2012.
  - [37] C. Chen, B. He, and X. Tang. Green-Aware Workload Scheduling in Geographically Distributed Datacenters. In Proc. 4th International Conference on Cloud Computing Technology and Science (CloudCom), pp 82–89, 2012.
  - [38] Y. Zhang, Y. Wang, and X. Wang. GreenWare: Greening Cloud-Scale Datacenters to Maximize the Use of Renewable Energy. In Proc. of ACM/IFIP/USENIX 12th International Middleware Conference, pp 143–164, 2011.
  - [39] H. Xu and B. Li. A General and Practical Datacenter Selection Framework for Cloud Services. In Proc. IEEE Fifth International Conference on Cloud Computing, pp 9–16, 2012.
  - [40] S. Ren, Y. He, and F. Xu. Provably-Efficient Job Scheduling for Energy and Fairness in Geographically Distributed Datacenters. In Proc.

- 32nd IEEE International Conference on Distributed Computing Systems, pp 22–31, 2012.
- [41] Z. Wu and J. Wang. Power Control by Distribution Tree with Classified Power Capping in Cloud Computing. In Proc. International Conference on Green Computing and Communications & International Conference on Cyber, Physical and Social Computing (CPSCoM), pp 319–324, 2010.
  - [42] X. Yang. Engineering Optimization – An Introduction with Metaheuristic Applications. John Wiley and Sons, New York, 2010.
  - [43] P. Venkataraman. Applied Optimization with MATLAB Programming. John Wiley and Sons, New York, 2002.
  - [44] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.
  - [45] R. M. Freund. Applied Lagrange Duality for Constrained Optimization. Massachusetts Institute of Technology, 2004.
  - [46] P. Y. Papalambros and D. J. Wilde. Principles of Optimal Design – Modeling and Computation, Second Edition, Cambridge University Press, 2000.
  - [47] United States Regional Electricity Prices, available online at www.eia.doe.gov.

Biographies

Rawan Y. Alawnah received her B.Sc. in Computer Engineering and a M.Sc. in Computer Engineering from Kuwait University, Kuwait, in 2010 and 2013, respectively. Earlier she worked as a teaching assistant in the Department of Computer Engineering at Kuwait University. Currently, she is a lab instructor in the Faculty of Engineering at American University of the Middle East, Kuwait. Her research interests include cloud computing and computer networks.

Imtiaz Ahmad received his B.Sc. in Electrical Engineering from University of Engineering and Technology, Lahore, Pakistan, a M.Sc. in Electrical Engineering from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, and a Ph.D. in Computer Engineering from Syracuse University, Syracuse, New York, in 1984, 1988 and 1992, respectively. Since September 1992, he has been with the Department of Electrical and Computer Engineering at Kuwait University, Kuwait, where he is currently a professor. His research interests include design automation of digital systems, high-level synthesis, and parallel and distributed computing.

Ebrahim A. Alrashed is an assistant professor in the Department of Computer Engineering at Kuwait University. His research interests include network security, mobile and wireless networks, cloud computing and sensor networks. He received his B.Sc. from University of the Pacific in 1988, both M.Sc. and Ph.D. in Computer Engineering from University of Southern California in 1993, and 1997, respectively.