Concurrent Placement, Capacity Provisioning, and Request Flow Control for a Distributed Cloud Infrastructure

Shuang Chen, Yanzhi Wang, Massoud Pedram
Department of Electrical Engineering
University of Southern California
Los Angeles, USA
{shuangc, yanzhiwa, pedram}@usc.edu

***Abstract*—Cloud computing and storage have attracted a lot of attention due to the ever increasing demand for reliable and cost-effective access to vast resources and services available on the Internet. Cloud services are typically hosted in a set of geographically distributed data centers, which we will call the cloud infrastructure. To minimize the total cost of ownership of this cloud infrastructure (which accounts for both the upfront capital cost and the operational cost of the infrastructure resources), the infrastructure owners/operators must do a careful planning of data center locations in the targeted service area (for example the US territories), data center capacity provisioning (i.e., the total CPU cycles per second that can be provided in each data center). In addition, they must have flow control policies that will distribute the incoming user requests to the available resources in the cloud infrastructure. This paper presents an approach for solving the unified problem of data center placement and provisioning, and request flow control in one shot. The solution technique is based on mathematical programming. Experimental results, using Google cluster data and placement/provisioning of up to eight data center sites demonstrate the cost savings of the proposed problem formulation and solution approach.**

I. Introduction

Cloud computing has been envisioned as the next-generation computing paradigm for its major advantages in on-demand self-service, ubiquitous network access, location independent resource pooling, and transference of risk [1]. In a cloud computing system, computation workloads are transferred from the users to the *cloud*, in which they are processed by data centers owned by the cloud service providers. These data centers are usually geographically distributed and may be mirror to each other for latency and reliability concerns. Typically, a data center occupies tens of thousands of square meters of land space, and consists of hundreds of thousands of computer servers, along with power delivery infrastructures, internal networking connections, and a cooling system. An overview of the architecture of a cloud computing system can be found in [2].

Given the specifications of data centers in a cloud infrastructure, the problem of request flow control and resource allocation has been well studied in a series of prior work. Authors of [3] developed a toolkit called "CloudSim" to simulate the cloud computing system and to evaluate the resource provisioning algorithm. Various models are used to analyze the behavior in a cloud computing system and to develop effective management schemes [4], [5].

On top of this resource allocation problem, there is the design problem of the placement and capacity provisioning of the data centers. Undersizing of data centers results in an increase of response time, degradation in quality of service, and finally loss of market for the service provider. And oversizing of the data centers increases the capital and operational cost with little practical use. When placing data centers, every major aspect of the capital and operational cost should be considered carefully because it is usually location dependent. For instance, Microsoft's data center in Quincy, Washington consumes 48MW of electricity power [6]. The electricity cost of this data center will be around 1.4 million dollars per month. However, to operate a data center with the same power consumption in Los Angeles, California, the electricity bill will increase to over 4 million dollars per month.

Some prior work have addressed the placement and/or the capacity provisioning problem of data centers. For instance, in [7], the authors identified the main costs in data centers and stated that their placement and provisioning can have a significant impact on service profitability. The authors of [8] provided a generalized cost calculation method and considered the availability of the cloud service. In [9], the authors addressed the problem of green computing and had a discussion on the tradeoff between the carbon footprint, the average cost, and the latency when different locations are selected to build new data centers.

While it is obvious that different placement and capacity provisioning schemes of the data centers result in different request flow control and resource allocation schemes, the request flow control and resource allocation algorithm can in turn affect how a placement and capacity provisioning performs. Due to this interdependency, any placement and capacity decision of the data centers should also be made based on realistic request routing and resource allocation. Unfortunately, the prior work has only applied very simple routing models and barely considered the problem of resource allocation inside a data center.

In this paper, we propose a generalized joint optimization framework of both data center placement/capacity provisioning and request flow control/resource allocation. The objective is to minimize the total average cost of the data centers in the network provided an average delay constraint. The request generation behavior of the cloud users is summarized based on the Google cluster data [10]. The problem of resource

978-3-9815370-2-4/DATE14/©2014 EDAA

allocation is formulated using the Generalized Processor Sharing (GPS) model [11], [12], in which the processor allocates a certain amount of its total computation resources to each request running on it. And the response time of a request is calculated accordingly. In the proposed model, we account for all the major aspects of the capital and operational cost of a data center, which can be time dependent and/or location dependent. In addition to building new data centers, we also allow the option of upgrading existing data centers. We provide an example based on the geographical information of the United States, and the optimal offline solution is presented.

The rest of the paper is organized as follows: In Section II, we introduce the system model we use. The problem formulation and the solution framework are presented in Section III. Section IV shows the experimental results. And the last section is the conclusion.

II. System model

A. User behavior

Let $\boldsymbol{U}$ denote the set of cloud users. Please note that when the problem we are interested in is in the scale of a large country such as the United States, it is neither feasible nor necessary to address to each individual user in the network. In such cases, we divide the whole country into small regions and aggregate the individual users in each region into one user node, corresponding to one element $u \in \boldsymbol{U}$. According to [13], the amount of user activity can be four times higher in peak hours than in off-peak hours. This means that the number of service requests may have significant fluctuation in one day. Therefore, a single request generation rate is not enough to fully characterize the user behavior. Generally, we can divide one day into a set of time periods, denoted by $\boldsymbol{H}$. For each time period $h \in \boldsymbol{H}$, the request generation rate of user u is denoted by λ_u^h. The total lasting time of time period h in one day is denoted by l_h. In the case that the users' behavior is similar and can be characterized by the same set of request generation rate in different time sections, we include these time sections into one time period for simplicity. In other words, each time period in our problem may be comprised of a set of disjoint time sections. A simple example would be dividing a whole day into peak hours, h_{peak}, and off-peak hours, $h_{\text{off-peak}}$. Apparently, $\lambda_u^{h_{\text{peak}}}$ should be greater than $\lambda_u^{h_{\text{off-peak}}}$.

B. Data center location and capacity

Let $\boldsymbol{D}$ denote the set of available locations for data centers. For each location $d \in \boldsymbol{D}$, let $y_d = 1$ if a data center is going to be built at location d, and $y_d = 0$ otherwise. For each location $d \in \boldsymbol{D}$, let $z_d = 1$ if a data center is already built at location d, and $z_d = 0$ otherwise. We assume that the servers in the data centers are homogeneous, and then the size of a data center can be characterized by the number of servers installed. Let n_d denote the number of servers in a new data center at location d if $y_d = 1$ or the number of servers added to an existing server if $z_d = 1$. Due to some reasons, e.g. limited land space or available electricity generation, we cannot install infinite number of servers in a data center. Also, it is not practical to build up a data center with just a few servers. The minimum and maximum numbers of servers that can be added to each location are denoted by $N_{d,\min}$ and $N_{d,\max}$, respectively. Alternatively, we can express the size of a data center in terms of its amount of computation resources. If the processing speed of one server is μ_s, then the total processing speed of a data center with n_d severs, denoted by μ_d, can be calculated as $\mu_d = n_d \cdot \mu_s$. In spite of the discrete nature, n_d and μ_d can be assumed to be able to take continuous values in the case that the total number of servers in a data center is large. For existing data centers, we denote the original number of servers and the original processing speed by $n_{0,d}$ and $\mu_{0,d}$, respectively. For new data centers, we define $n_{0,d} = 0$ and $\mu_{0,d} = 0$.

C. Routing and delay modeling

As described in Section II.A, the users behave differently in different time periods. Therefore, the routing strategy should be specified accordingly. Let $x_{u,d}^h = 1$ if during time period h, some of the requests generated by user u are routed to the data center at location d, and $x_{u,d}^h = 0$ otherwise. And let $\lambda_{u,d}^h$ denote the rate that requests are routed from user u to the data center at location d during time period h when $x_{u,d}^h = 1$. According to the GPS model, the average processing latency of these requests, $t_{proc,u,d}^h$, can be calculated as

$$t_{proc,u,d}^h = \frac{1}{\phi_{d,u}^h \cdot (\mu_d + \mu_{0,d}) - \lambda_{u,d}^h} \tag{1}$$

where $\phi_{d,u}^h$ is the proportion of computation resources that data center at location d allocated to user u during time period h. The equivalent processing speed for requests routed from user u to data center at location d, denoted by $\psi_{d,u}^h$, can be calculated as

$$\psi_{d,u}^h = \phi_{d,u}^h \cdot (\mu_d + \mu_{0,d}) \tag{2}$$

It reflects the processing capability seen from a user's side as if it is exclusively used for the user.

Let $t_{prop,u,d}$ and $t_{prop,d,u}$ denote the propagation delay from user u to location d and the propagation delay from location d to user u, respectively. Then the average total delay experienced by a request routed from user u to the data center at location d during time period h, denoted by $t_{total,u,d}^h$, can be calculated as:

$$t_{total,u,d}^h = t_{prop,u,d} + t_{proc,u,d}^h + t_{prop,d,u} \tag{3}$$

where $t_{proc,u,d}^h$ is calculated as in (1). Generally, the propagation delays in the two directions are not necessarily the same because the data packets may go through different paths in the network. However, since the analysis of all the routers' behavior in the network is complicated and out of the scope of this paper, we use the estimation of propagation delay based on Euclidean distance between the source and the destination, in which case the propagation delay in two directions will be the same.

D. Power consumption modeling

For a data center consisting of homogeneous servers, let P_{idle} and P_{peak} denote the power consumptions of a server at the utilization level of 0 and 100%, respectively. And let $E_{usage,d}$ denote the power usage effectiveness (PUE) [14] of a data center at location d. Then according to [15], the total power consumption of a data center at location d during time period h, denoted by $P_{total,d}^h$, can be calculated as

$$\begin{aligned} P_{total,d}^h =& (n_d + n_{0,d}) \cdot [P_{idle} + (E_{usage,d} - 1) \cdot P_{peak}] \\ &+ (n_d + n_{0,d}) \cdot (P_{peak} - P_{idle}) \cdot \gamma_d^h + \epsilon \end{aligned} \tag{4}$$

where γ_d^h is the utilization level of the data center during time period h, which can be obtained by

$$\gamma_d^h = \left(\sum_u x_{u,d}^h \lambda_{u,d}^h \right) / (\mu_d + \mu_{0,d}) \tag{5}$$

and ϵ is an empirical constant. If the data center treats all the servers in it fairly, then the power consumption corresponding to the newly installed servers, denoted by $P_{new,d}^h$, can be approximated as

$$\begin{aligned} P_{new,d}^h = &\{n_d[P_{idle} + (E_{usage,d} - 1)P_{peak}] \\ &+ n_d(P_{peak} - P_{idle})\gamma_d^h + \epsilon \end{aligned} \tag{6}$$

Please note that $P_{total,d}^h$ is equal to $P_{new,d}^h$ for a new data center since $n_{0,d} = 0$.

Define $P_{total,d}$ as the maximum power consumption of the data center at location d. It can be calculated by applying $\gamma_d^h = 1$ to (4) as

$$P_{total,d} = (n_d + n_{0,d}) \cdot E_{usage,d} \cdot P_{peak} + \epsilon \tag{7}$$

Similarly, we define $P_{new,d}$ the maximum power consumption of all the new servers. $P_{new,d}$ can be obtained by

$$P_{new,d} = n_d \cdot E_{usage,d} \cdot P_{peak} + \epsilon \tag{8}$$

These two parameters are useful for the estimation of the cost of a data center, which is introduced in Section III.A.

PUE is a parameter that characterize the power consumption of components other than the servers. It is related to the geographical location. For instance, PUE is usually higher in regions with higher temperature or humidity because the cooling system consumes more energy in these regions.

III. Problem formulation and solution method

We formulate our problem as an optimization problem with the objective to minimize the average total cost of all the new and existing data centers in the network subject to the constraint of a maximum allowable average latency for each user. We first calculate the overall cost of a data center.

A. Cost calculation

The overall cost of a data center over its lifetime is comprised of various aspects of capital cost and operational cost, each of which may depend on the time, the size of the data center and/or the location of the data center. We will address these aspects separately.

Land cost. The land cost, denoted by $C_{land,d}$, is the cost of buying enough land space to build up a data center, and can be calculated as follows

$$C_{land,d} = LandPrice(d) \cdot Area(d) \tag{9}$$

where $LandPrice(d)$ is the cost of unit area at location d, and $Area(d)$ is the area required for a data center at location d. According to [16], the required area of a data center can be estimated as linearly proportional to its maximum total power consumption. Since we do not need to account for the land space occupied by the original part of an existing data center, $P_{new,d}$ is used instead of $P_{total,d}$. (9) can be rewritten as

$$C_{land,d} = LandPrice(d) \cdot k_{p-a} \cdot P_{new,d} \tag{10}$$

where k_{p-a} is the ratio between the required area of a data center and its maximum total power consumption.

Infrastructure cost and server cost. The infrastructure cost, denoted by $C_{infra,d}$, is the cost of installing the power delivery network, the cooling system, and the internal networking within a data center. Similar to the land cost, the infrastructure cost of a data center can be estimated as

$$C_{infra,d} = k_{p-infra} \cdot P_{new,d} \tag{11}$$

where $k_{p-infra}$, usually in the unit of \$/MW, is the infrastructure cost per unit power consumption of the data center. The server cost, denoted by $C_{serv,d}$, is the cost of buying new servers for a data center and can be straightforwardly calculated as

$$C_{serv,d} = n_d \cdot ServerPrice \tag{12}$$

where $ServerPrice$ is the price of a single server.

Connection cost. The connection cost, denoted by $C_{conn,d}$, is the cost to lay out the optical fibers to the nearest Internet Backbone and the cost to lay out power transmission lines to the existing electric power network, and can be calculated as

$$\begin{aligned} C_{conn,d} = &TransLinePrice \cdot DistPower(d) \\ &+ FiberPrice \cdot DistInternet(d) \end{aligned} \tag{13}$$

where $DistPower(d)$ is the distance from the data center at location d to the nearest power plant or the existing transmission line, $DistInternet(d)$ is the distance from the data center at location d to the nearest Internet backbone, and $TransLinePrice$ and $FiberPrice$ are the prices of transmission line and optical fiber per unit length, respectively. For an existing data center, we do not have this part of cost since the connection is already built up along with the data center.

Cooling cost. The cooling cost of a data center can be divided into two parts. One is the cost of electricity consumed by the computer room air conditioners (CRACs), which is characterized by the PUE and covered in the electricity cost. And the other is the cost of water wasted in the cooling circulation, which is specified in the Water cost.

Electricity cost. The electricity cost, denoted by $C_{elec,d}^h$, is the cost of the electricity energy consumed while operating the data center. Since the total electricity power consumption of a data center can be obtained using (4), the calculation of $C_{elec,d}^h$ is straightforward.

$$C_{elec,d}^h = ElectricityPrice(d) \cdot P_{total,d}^h \tag{14}$$

where $ElectricityPrice(d)$ is the price of unit electricity energy at location d.

Bandwidth cost. The bandwidth cost, denoted by $C_{band,d}$, is the cost of acquiring sufficient bandwidth for communications from the internet service providers (ISPs). If we allocate constant bandwidth to each server in the data center, then $C_{band,d}$ can be calculated as

$$C_{band,d} = BWPrice \cdot (n_d + n_{0,d}) \cdot BWServer \tag{15}$$

where $BWPrice$ is the price of unit amount of bandwidth set by the ISPs, and $BWServer$ is the amount of bandwidth allocated to each server.

Water cost. As introduced in the description of the cooling cost, the water cost, denoted by $C_{water,d}$, is the cost of the water wasted by the water chiller, which can be estimated

from the maximum total power consumption of the data center. $C_{water,d}$ is calculated as

$$C_{water,d} = WaterPrice(d) \cdot k_{p-w} \cdot P_{total,d} \quad (16)$$

where $WaterPrice(d)$ is the price of unit amount of water at location d, and k_{p-w} is the amount of water required by unit of maximum power consumption.

Maintenance cost. The maintenance cost, denoted by $C_{main,d}$, is the cost of hiring personnel to maintain and operate a data center and can be calculated as follows

$$C_{main,d} = k_{p-main} \cdot P_{total,d} + Salary(d) \cdot (n_d + n_{0,d})/k_{s-p} \quad (17)$$

where the first term on the right-hand side is the maintenance cost and the second term on the right-hand size is the operational cost. k_{p-main} is the ratio between the maintenance cost and the maximum power consumption, $Salary(d)$ is the salary to hire an administrator at location d, and k_{s-p} describes how many servers one administrator can manage.

To obtain the overall cost of a data center in one day, the one-time investment including the land cost, the infrastructure cost, the server cost, and the connection cost should be amortized over the expected life time of the data center, a server, the transmission line, or the optical fiber. Therefore, if $y_d = 1$, i.e. we will build a new data center at location d, then the amortized overall cost per day of this data center, denoted by $C_{total,d}$, can be calculated as

$$\begin{aligned} C_{total,d} =& \frac{C_{land,d}}{T_{dc}} + \frac{C_{infra,d}}{T_{dc}} + \frac{C_{serv,d}}{T_{serv}} + \frac{C_{conn,d}}{T_{line}} \\ &+ \sum_h l_h \cdot C^h_{elec,d} + C_{band,d} + C_{water,d} + C_{main,d} \end{aligned} \quad (18)$$

where T_{dc}, T_{serv}, and T_{line} are the expected lifetimes for the corresponding components.

For an existing data center at location d with $z_d = 1$, all the cost calculation is the same except that the connection cost, $C_{conn,d}$, is set to zero.

B. Problem formulation

The average latency is calculated as the weighted average of the latency values of requests routed to each data center. Hence, in the case that the requests from one user node are routed to multiple data centers with different average latencies, the overall average latency can still satisfy the average latency constraint even if a small portion of requests routed to some data centers experience extremely long latency. However, we eliminate this possibility by enforcing the latency constraint on every routing path because it can cause serious unfairness otherwise. This is because each user node in our problem consists of a large number of individual users, and the request from some individual users may be consistently routed to the data centers with long latency due to the implemented routing scheme in reality.

Based on the assumption of homogeneous servers, we can combine (4) and (5), and get

$$\begin{aligned} P^h_{total,d} =& (n_d + n_{0,d}) \cdot [P_{idle} + (E_{usage,d} - 1) \cdot P_{peak}] \\ &+ \mu_s \cdot (P_{peak} - P_{idle}) \cdot \sum_u x^h_{u,d} \lambda^h_{u,d} + \epsilon \end{aligned} \quad (19)$$

From (9) to (18), one can see that the total cost of a data center is a monotonically increasing function of the maximum power consumption of the data center. And from (19), one can see that the maximum power consumption is a non-decreasing function of the number of servers in the data center. Therefore, there is no reason to install extra servers to have to a data center with higher processing speed once the average latency constraint is satisfied. This being true, we can transform the constraint of maximum average latency to the required equivalent processing speed. Combining (1) with (3), we get

$$\psi^h_{d,u} = 1/(t^h_{u,MAX} - t_{prop,u,d} - t_{prop,d,u}) + \lambda^h_{u,d} \quad (20)$$

where $t^h_{u,MAX}$ is the maximum allowable average latency for user u during time period h.

Given the values of z_d's, λ^h_u's, $n_{0,d}$'s, N_d's, μ_s, and all other information required during the cost calculation, the optimization problem can be formulated as follows:

Find $x^h_{u,d}, y_d, n_d, \lambda^h_{u,d}$, $\forall u \in \boldsymbol{U}, \forall d \in \boldsymbol{D}, \forall h \in \boldsymbol{H}$

Minimize

$$\sum_{d \in \boldsymbol{D}} (y_d + z_d) C_{total,d} \quad (21)$$

Subject to

$$\sum_d \lambda^h_{u,d} = \lambda^h_u, \qquad \forall u, h \quad (22)$$

$$\sum_u x^h_{u,d} \psi^h_{d,u} \leqslant (n_{0,d} + n_d)\mu_s, \qquad \forall d, h \quad (23)$$

$$y_d \leqslant 1 - z_d, \qquad \forall d \quad (24)$$

$$\frac{\lambda^h_{u,d}}{\lambda^h_u} \leqslant x^h_{u,d}, \qquad \forall u, d, h \quad (25)$$

$$x^h_{u,d} \leqslant x^h_{th,u,d} - \delta, \qquad \forall u, d, h \quad (26)$$

$$\frac{n_d}{N_d} \leqslant y_d + z_d, \qquad \forall d \quad (27)$$

$$x^h_{u,d} \leqslant y_d + z_d, \qquad \forall u, d, h \quad (28)$$

$$x^h_{u,d} \in \{0, 1\}, \qquad \forall u, d, h \quad (29)$$

$$y_d \in \{0, 1\}, \qquad \forall d \quad (30)$$

$$n_d \in [N_{d,\min}, N_{d,\max}] \cap \mathbb{N}, \qquad \forall d \quad (31)$$

$$\lambda^h_{u,d} \geqslant 0, \qquad \forall u, d, h \quad (32)$$

where $C_{total,d}$ is specified in (18), $\psi^h_{d,u}$ is calculated in (20), δ is set to a small positive value for the convenience of optimization, and $x^h_{th,u,d}$ is defined as

$$x^h_{th,u,d} = \frac{t^h_{u,MAX} - t_{prop,u,d} - t_{prop,d,u}}{t^h_{u,MAX} + t_{prop,u,d} + t_{prop,d,u}} + 1 \quad (33)$$

$x^h_{th,u,d}$ being less than or equal to 1 means that the propagation delay between the user and the data center is at least the maximum allowable average latency, and no matter how much computation resource is allocated to this portion of request, the delay constraint will be violated.

The objective function is the sum of the total cost of all the data centers over 24 hours of a day. (22) ensures that all the user requests at any time are routed to data centers for processing. (23) ensures that the allocated computation resources in each data center do not exceed its maximum amount of available computation resources. We have (24) because the locations of existing data centers are not the candidate locations for building new data centers. We force

$\lambda^h_{u,d}$ to be 0 when we do not choose to route any request from user u to the data center at location d during time period h by (25). With (26), no requests are routed to those data centers that cannot respond within the latency constraint. With (27), no servers are installed where no data center already exists or is to be built. And with (28), no requests are routed to non-existing data centers. (29)(30)(31)(32) specify the ranges of value for variables $x^h_{u,d}$, y_d, n_d, and $\lambda^h_{u,d}$.

C. Solution method

Generally, this is a mixed integer non-linear programming (MINLP) problem. Even though MINLP problems are NP-hard, it is acceptable to solve the problem directly with some standard solvers, such as CPLEX [17], regardless of the computational complexity, since the problem is solved offline for only once. Nevertheless, some simplification can be made to the original problem. As is stated in Section II.B, since the number of servers in a data center is considerably large, we can assume that n_d's can take continuous values with negligible error in cost and delay calculation, which reduce the number of integer variables in the problem. Moreover, once the values of $x^h_{u,d}$'s and y_d's are given, the objective function and all the constraints are linear. Therefore, we can apply stochastic optimization methods such as simulated annealing [18] or genetic algorithm [19] combined with linear programming to solve the problem.

We also propose a greedy algorithm for sub-optimal solutions, which is applied in the following steps. Initially, we set all the y_d's to 1 and all the n_d's to 0. For each time period h, we first pick one user u and set the routing scheme to the one with the minimum cost increase among the routing schemes in which all the requests are routed to a single data center. This process is repeated for every user. When the routing for all the users in every time period is done, we can find out the minimum capacity for each data center, and the data center that is not used will not be built. In the simple case with uniform user behavior over time, no existing data centers and ϵ in (4) equal to 0, if the data center can be sized arbitrarily and all the candidate location can be connected to the power network and Internet with minimal cost, then the greedy algorithm can be proved to solve the problem optimally.

IV. Experimental results

In this section, we give an example on a service area of the United States.

A. Experimental setup

We set the ten most populous cities in the U.S. as the user nodes. The population information is obtained from the U.S. Census Bureau website[1]. We select to build the data centers among eight candidate locations, which are Austin, Bismarck, Los Angeles, New York City, Oklahoma City, Orlando, Seattle, and St. Louis. The World Geodetic System (WGS) coordinates of the user nodes and the candidate building locations are obtained from the GeoNames database[2]. The propagation delay between a data center and a user node is set to be linearly proportional to the distance between the two locations. The propagation delay increases by 5ms for every 100km of distance. For each user and time period, the maximum average latency, $t^h_{u,MAX}$, is set to t_{MAX} uniformly.

Each day is divided into peak hours and off-peak hours. Through the Google cluster data, we obtain the request arrival pattern by averaging the number of requests arriving in each hour within a period of one month. We then divide the 24 hours of a day into 12 peak hours and 12 off-peak hours and calculate the ratio between the average request arrival rates of peak hours and off-peak hours. We define this ratio to be also the ratio of request generation rate in each user node during the peak hours and the off-peak hours. We assume a 1 request per second generation rate per 40,000 population in each user node during the peak hours. And the request generation rate during the off-peak hours can be calculated accordingly.

For each data center, the minimum and maximum number of servers are set to 1,000 and 50,000, respectively. We use Dell PowerEdge R610 as the server model, which has a peak power consumption of 260W and an idle power consumption of approximately 160W. Each server by itself is assumed to process requests with an average response time of 100s. The empirical constant, ϵ, is set to 0. The PUE of a data center is modeled as a function of the temperature as in [8].

Other parameters in the cost calculation are specified as follows. The land price for each location is calculated by averaging the prices looked up on real estate websites. k_{p-a} in (10) is set to 6,000 sf/MW as recommended in [16]. $k_{p-infra}$ in (11) is set to \$15/W [16]. Each server costs \$2,000. The electric power grid map is obtained from the National Public Radio website[3] and each candidate city is covered by the existing power transmission lines. The Internet backbone maps can also be found online[4], and the cost of optical fiber is set to \$480,000 per mile. Each server is allocated 1Mbps of network bandwidth, and the price of bandwidth is set to \$1/Mbps. The electric power prices in different states are obtained from the U.S. Energy Information Administration website[5]. The water prices are obtained from the government websites of each city, and k_{p-w} in (16) is set to 24,000 gal/MW/day. The maintenance cost is set to \$0.05/W/month [20]. The salary of each administrator is set to \$100,000 per year, and k_{s-p} in (17) is set to 1,000. The expected lifetime of a data center, a server, and the optical fiber is set to 12 years, 4 years, and 12 years, respectively.

B. Simulation results

In this part, we will present the simulation results under two different scenarios.

In Scenario 1, there are no existing data centers, i.e. $z_d = 0, \forall d$. Two baselines are chosen by randomly place three data centers among the eight candidate locations. In Baseline 1, data centers are built in Los Angeles, New York City, and Seattle. In Baseline 2, data centers are built in Bismarck, Oklahoma City, and St. Louis. The relationship between the overall cost per day and t_{MAX} is shown in Fig. 1. The cost of Baseline 2 when $t_{MAX} = 100ms$ is not shown because the latency constraint cannot be satisfied in that case. Generally, the placement and capacity of the data centers are different with different t_{MAX} in our result. For example, when $t_{MAX} = 200ms$, data centers are built in Oklahoma City and St. Louis with 46,671 servers and 23,322 servers, respectively. When $t_{MAX} = 600ms$, data centers are built in Austin, Oklahoma City, and Seattle with 15,741 servers,

[1] http://www.census.gov/2010census/popmap/

[2] http://www.geonames.org/

[3] http://www.npr.org/templates/story/story.php?storyId=110997398

[4] http://www.nthelp.com/maps.htm

[5] http://www.eia.gov/electricity/

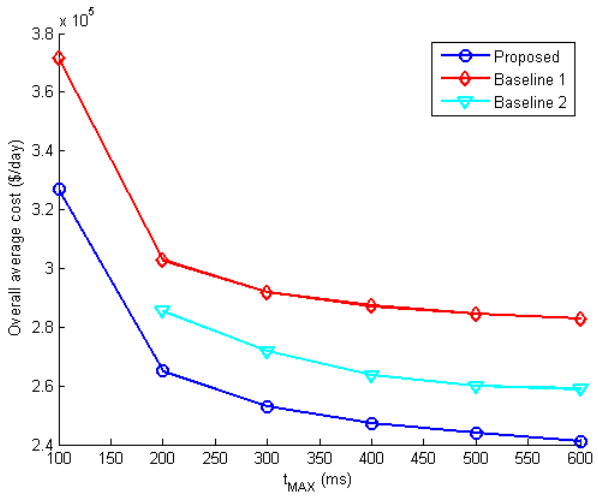

Fig. 1. Simulation Result of Scenario 1

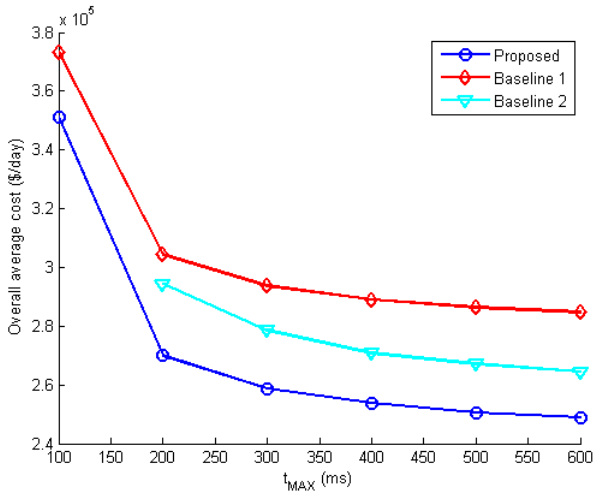

Fig. 2. Simulation Result of Scenario 2

38,094 servers, and 9,158 servers, respectively. One can make the observation that the overall cost is lower with higher t_{MAX}, which is consistent with the fact that looser constraints result in better performance. It can also be seen from the result that the overall cost of the proposed schemes always lower than that of the baselines. When $t_{MAX} = 500ms$, the proposed method saves \$1.22M and \$475k per month compared to Baseline 1 and Baseline 2, respectively. In Scenario 2, there already exists a data center in Seattle with 5,000 servers. The same baseline schemes are used for comparison. The result is shown in Fig. 2. As can be seen from the figure, the result is similar to that of Scenario 1.

V. Conclusion

In this paper, we proposed a joint optimization framework of request flow control/resource allocation and data center placement/capacity provisioning in a cloud cloud infrastructure. The major aspects of the capital and operational cost of a data center are accounted for. An optimization problem is formulated with the objective to minimize the overall cost of all the data centers in the network and a constraint on the maximum average latency for each service request. Simulation results are presented to show that the proposed scheme can achieve considerable cost saving over the baseline schemes.

Acknowledgment

This work is supported in part by the Software and Hardware Foundations program of the NSFs Directorate for Computer & Information Science & Engineering.

References

[1] B. Hayes, "Cloud computing," *Commun. ACM*, vol. 51, no. 7, pp. 9–11, Jul. 2008.

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation computer systems*, vol. 25, no. 6, pp. 599–616, 2009.

[3] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[4] H. Goudarzi and M. Pedram, "Multi-dimensional sla-based resource allocation for multi-tier cloud computing systems," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. IEEE, 2011, pp. 324–331.

[5] Y. Wang, S. Chen, H. Goudarzi, and M. Pedram, "Resource allocation and consolidation in a multi-core server cluster using a markov decision process model." in *ISQED*, 2013, pp. 635–642.

[6] R. Katz, "Tech titans building boom," *Spectrum, IEEE*, vol. 46, no. 2, pp. 40–54, 2009.

[7] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 68–73, 2008.

[8] I. Goiri, K. Le, J. Guitart, J. Torres, and R. Bianchini, "Intelligent placement of datacenters for internet services," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, 2011, pp. 131–142.

[9] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Energy-information transmission tradeoff in green cloud computing," *Carbon*, vol. 100, p. 200, 2010.

[10] "Google cluster data." [Online]. Available: https://code.google.com/p/googleclusterdata/

[11] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Transactions on Networking (TON)*, vol. 1, no. 3, pp. 344–357, 1993.

[12] Z.-L. Zhang, D. Towsley, and J. Kurose, "Statistical analysis of generalized processor sharing scheduling discipline," in *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4. ACM, 1994, pp. 68–77.

[13] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 123–134, 2009.

[14] R. Brown *et al.*, "Report to congress on server and data center energy efficiency: Public law 109-431," 2008.

[15] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 13–23, 2007.

[16] W. P. Turner and J. H. Seader, "Dollars per kw plus dollars per square foot are a better datacenter cost model than dollars per square foot alone," *Uptime Institute White Paper*, 2006.

[17] I. CPLEX, "11.0 users manual," *ILOG SA, Gentilly, France*, 2007.

[18] P. J. Van Laarhoven and E. H. Aarts, *Simulated annealing*. Springer, 1987.

[19] D. Whitley, "A genetic algorithm tutorial," *Statistics and computing*, vol. 4, no. 2, pp. 65–85, 1994.

[20] L. A. Barroso and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis Lectures on Computer Architecture*, vol. 4, no. 1, pp. 1–108, 2009.