Optimal Resource Allocation for Multimedia Application Providers in Multi-site Cloud

Xiaoming Nan, Yifeng He and Ling Guan
Department of Electrical and Computer Engineering, Ryerson University, Toronto, Ontario, Canada

Abstract—Cloud-based multimedia applications have been widely used in recent years. With the advance of globalization, application providers hope to offer services at multiple sites serving users all over the world. The key challenge is how to effectively manage resource allocation and workload balancing. In this paper, we study the optimal resource allocation problem in multi-site cloud. Specifically, we jointly optimize the global workload assignment and the local VM allocation to minimize the resource cost under the service response time requirements. Moreover, we propose a greedy algorithm to efficiently apply our study in a practical way. Experimental results demonstrate that the proposed optimal resource allocation scheme can optimally allocate resources and balance workload to achieve the minimal resource cost for multimedia application providers.

I. INTRODUCTION

In recent years, we have witnessed the fast development of cloud computing. In cloud data centers, a shared pool of servers are managed to provide on-demand resources (e.g., computation, storage, platform, software, etc.) as services via the Internet. According to different levels of service provisioning, cloud computing can be categorized into Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), among which the SaaS model is most familiar to individual users. In the SaaS model, application providers rent VMs from cloud vendors and deliver services to users. By using cloud-based applications, users are free from the burden of application installation and software maintenance.

The rapid development of cloud computing has offered great opportunities for multimedia applications. Cloud-based multimedia applications, like on-line photo sharing, cloud-based video streaming, and various social media applications, have been increasingly used in daily life. For multimedia application providers, they have two major concerns: the Quality of Service (QoS) and the resource cost. Considering the delay-sensitive characteristic, the service response time is taken as the major QoS factor. The service response time is defined as the duration from the time when a request arrives at a data center to the time when the requested application has been completely served. The service response time is contracted into the Service Level Agreement (SLA). Thus, it is important to meet service response time requirements for all users. Besides QoS, the resource cost is another important concern. Generally, cloud vendors can offer two different VM rental schemes: the reservation scheme and the on-demand scheme. Price rates in the reservation scheme are much lower than those in the on-demand scheme. But VMs in the reservation scheme have to be subscribed in advance with deposit. During the service provisioning, if the initially reserved VMs cannot meet resource demands, application providers can request on-demand VMs instantly at the expense of a higher price rate. Facing different applications and resource demands, it is significant for application providers to provide satisfactory services at a modest cost.

With the advance of globalization, application providers hope to provide services to different locations. They are also seeking to distribute workload among data centers in order to efficiently utilize resources and increase the availability of applications. Owing to the increasing scale of cloud based applications, an effective resource allocation is becoming one of the biggest challenges faced by application providers. Currently, the common approaches are the distributed approaches. In the distributed approaches, a local controller gathers runtime information, balances workload, and manages resource allocation in the data center. The distributed approaches are effective for one specific data center, but they lack an overview of the whole system, probably leading to the unbalanced workload and even the local congestion. For example, some data centers need extra on-demand VMs to tackle the increasing workload while some other data centers still have spared VMs. Lacking global information, the distributed approaches in multi-site cloud can only achieve a local optimum.

To address above mentioned challenges, we study the resource allocation problem in this paper. Our contribution can be presented as follows. We propose the optimal resource allocation scheme for multimedia application providers, in which we jointly optimize the global workload assignment and the local VM allocation to minimize the resource cost under the service response time requirements. Since the formulated optimization problem is a NP-hard problem, we also propose a greedy algorithm to efficiently allocate VMs in a practical way. Experimental results demonstrate that the proposed resource allocation scheme can effectively utilize resources and balance workload to achieve a minimal resource cost.

II. RELATED WORK

The resource allocation in cloud has always been a challenging research topic. Nan et al. [1] propose queueing model based resource allocation schemes for multimedia cloud. Chaisiri et al. [2] present an optimal VM placement algorithm based on the stochastic integer programming. However, [1]
and [2] fail to consider resource allocation in multiple data centers. Bouyoucef et al. [3] formulate virtual server allocation as a linear programming. Nan et al. [4] propose distributed approaches to optimize VM allocation. But [3] and [4] lack an overview of the whole system, leading to a local optimum. Comparing with the work in [1-4], our paper is different in the following senses: 1) we jointly optimize the global workload and the local resource allocation; 2) we provide the theoretical optimal solution and the practical greedy algorithm.

III. SYSTEM MODELS
A. SaaS Architecture in Multi-site Cloud

In multi-site cloud, application providers can deploy services at multiple data centers. Each data center consists of a bunch of servers to support the running and provisioning of VMs. Generally, different classes of VM instances have different configuration levels in terms of compute unit, processor frequency, memory size, I/O rate, etc. Multiple VMs can compose virtual cluster to accelerate processing speed.

The proposed SaaS architecture is illustrated in Fig. 1. The service process can be summarized as follows. When requests arrive at data center, the dispatcher will schedule requests to the corresponding request queue. The workload monitor performs a live monitoring on the type and number of requests, and reports to the local controller. The local controller takes two responsibilities. Firstly, it gathers information including local workload and available resources, and forwards the information to the global controller for analysis. Secondly, the local controller executes commands from the global controller, transfers workload, and allocates VMs in the data center. Based on the workload and resource information from all local controllers, the global controller jointly optimizes the workload assignment and VMs allocation to minimize the total resource cost.

B. Resource Allocation Model

We propose the resource allocation model to study the optimal resource allocation in multi-site cloud. Since the workload in cloud computing is time varying, we divide the time domain into time slots. We choose a time slot \( t \) which is short enough so that the workload in each time slot \( t \) is constant. Suppose that \( L \) data centers are distributed at different locations and the set of data centers can be denoted as \( D = \{ D_1, D_2, \ldots, D_L \} \). At time slot \( t \), the transmission delay between \( D_i \) and \( D_j \) is denoted as \( d_{ij}^t \). Specially, when \( l\' = l, d_{ll}^t = 0 \). Suppose that \( N \) classes of VMs are provided. Let \( p_{ij}^t \) and \( p_{ij}^d \) denote the price rates of reserved and on-demand class-\( i \) VM instance at \( D_l \), respectively. In the initial reservation phase, application providers will reserve a certain number of VMs in different classes. Let \( K_{li}^{ini} \) be the number of initially reserved class-\( i \) VMs at \( D_l \).

Suppose that there are \( M \) types of multimedia applications. For each application, different classes of VMs have different processing speed. Let \( \mu_{ij} \) be the service rate of class-\( i \) VM instance for processing type-\( j \) application requests. Since the two consecutive requests arriving at \( D_l \) may be from two different users, the inter-arrival time can be modeled as an exponential random variable [5]. Therefore, the initial arrivals of type-\( j \) application requests at \( D_l \) follow a Poisson distribution with an average of \( \lambda_{lj}^{ini} \). After global workload balancing, some type-\( j \) requests may be transferred to other data centers for service. Let \( \chi_{lj}^{t} \) denote the ratio of type-\( j \) requests transferred from \( D_l \) to \( D_{l'} \). Since all requests will be processed, we can get \( \sum_{j=1}^{M} \chi_{lj}^{t} = 1 \). With the requests from other data centers, the arrivals of type-\( j \) application requests at \( D_l \) can be formulated as \( \lambda_{lj}^{t} = \sum_{l'=1}^{L} \chi_{lj}^{t} \lambda_{lj}^{ini} \).

For type-\( j \) application at \( D_l \), let \( K_{lj}^{r(t)} \) and \( K_{lj}^{d(t)} \) denote the number of allocated class-\( i \) VMs in the reservation scheme and the on-demand scheme, respectively. We will propose the optimal resource allocation scheme to determine the optimal workload assignment \( \chi_{lj}^{t} \) and the optimal VMs allocation \( K_{lj}^{r(t)} \) and \( K_{lj}^{d(t)} \) (\( l \in \{1, \ldots, L\}, i \in \{1, \ldots, N\}, \) and \( j \in \{1, \ldots, M\} \)).

IV. OPTIMAL RESOURCE ALLOCATION SCHEME IN MULTI-SITE CLOUD

In this section, we study the optimal resource allocation scheme in multi-site cloud. Our objective is to minimize the total resource cost. The total resource cost can be formulated as \( C_{gl}^{t} = \sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} p_{ij}^{t} K_{lj}^{r(t)} + \sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} p_{ij}^{d} K_{lj}^{d(t)} \) in which the first term is the cost of VMs in the reservation scheme and the second term is the cost of VMs in the on-demand scheme.

We take the service response time as the QoS measurement. Let \( T_{lj}^{resp(t)} \) be the service response time of type-\( j \) application at \( D_l \). The service response time includes the local execution time \( T_{lj}^{exe(t)} \) and the requests transferring time \( T_{lj}^{tra(t)} \), i.e. \( T_{lj}^{resp(t)} = T_{lj}^{exe(t)} + T_{lj}^{tra(t)} \).
At $D_l$, $(K_{lij}^{r(t)} + K_{lij}^{d(t)})$ class-$i$ VMs work together as the class-$i$ virtual cluster to process type-$j$ requests with the service rate of $(K_{lij}^{r(t)} + K_{lij}^{d(t)}) \mu_{ij}$. To balance the workload among different virtual clusters, we use the normalized service rate as the scheduling probability, i.e., one type-$j$ request is scheduled to class-$i$ virtual cluster with the probability of $\omega_{ij}(t) = \frac{(K_{lij}^{r(t)} + K_{lij}^{d(t)}) \mu_{ij}}{\sum_{i=1}^{N}(K_{lij}^{r(t)} + K_{lij}^{d(t)}) \mu_{ij}}$. According to the decomposition property of Poisson Process, the arrivals of scheduled requests at class-$i$ virtual cluster also follow a Poisson Process with an average of $\omega_{ij}(t) \tau$. From [5], the processing time at class-$i$ virtual cluster can be approximated as exponential distribution with an average of $\frac{1}{\mu_{ij} \omega_{ij}(t)}$. Thus, the service process of type-$j$ application at class-$i$ virtual cluster can be analyzed as an $M/M/1$ queueing system [5]. To make the queue stable, $\omega_{ij}(t) \lambda_j < (K_{lij}^{r(t)} + K_{lij}^{d(t)}) \mu_{ij}$ should be satisfied. Therefore, the mean execution time of the type-$j$ application at $D_l$ is given by $T_{ij}^{exec}(t) = \sum_{i=1}^{N} \frac{(K_{lij}^{r(t)} + K_{lij}^{d(t)}) \mu_{ij} - \omega_{ij}(t) \lambda_j}{\mu_{ij}}$.

Since the workload can be dynamically redirected to other sites for service, we have to consider the transmission delay. Let $\Psi(t) = \{t \mid \exists z_t \neq 0\}$ denote the set of data centers which transfer type-$j$ requests towards $D_l$. In order to meet the QoS demands for all users, we choose the maximal transmission delay $\max \{d_{ij}^t(t) \mid l' \in \Psi(t)\}$ as the transferring time at $D_l$. If requests from the furthest data center can be satisfied, other requests’ demands will be satisfied. Thus, the transferring time $T_{ij}^{trans}(t)$ is given by $T_{ij}^{trans}(t) = \max \{d_{ij}^t(t) \mid l' \in \Psi(t)\}$.

Based on the above analysis, the service response time $T_{ij}^{resp}(t)$ can be formulated as $T_{ij}^{resp}(t) = T_{ij}^{exec}(t) + T_{ij}^{trans}(t) = \sum_{i=1}^{N} \frac{(K_{lij}^{r(t)} + K_{lij}^{d(t)}) \mu_{ij} - \omega_{ij}(t) \lambda_j}{\mu_{ij}} + \max \{d_{ij}^t(t) \mid l' \in \Psi(t)\}$.

At each data center, the total number of allocated class-$i$ VMs in the reservation scheme should be no more than the number of initially reserved class-$i$ VMs. The constraints can be formulated as $\sum_{j=1}^{K_{li}^{init}} K_{lij}^{r(t)} \leq K_{li}^{init}$. Based on the above analysis, we can state the optimal resource allocation problem in multi-site cloud as follows.

Minimize $\sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} p_{li} K_{lij}^{r(t)}$ subject to

$$\sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} p_{li} K_{lij}^{d(t)} + \sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} d_{ij}^t(t) \leq \tau_j$$

$$\sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} K_{lij}^{r(t)} \leq K_{li}^{init}$$

$$\sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} d_{ij}^t(t) \leq \tau_j$$

where $\tau_j$ is the upper bound of the service response time for type-$j$ application, $l \in \{1, \ldots, L\}$, $i \in \{1, \ldots, N\}$, and $j \in \{1, \ldots, M\}$.

The optimal resource allocation problem (1) is a mixed integer programming, which is known to be NP-hard. The problem can be solved by the branch-and-bound method [6]. However, since application providers require an efficient resource allocation scheme, which can be rapidly executed to adapt to the time varying workload. Therefore, we propose a greedy algorithm to allocate VMs in a practical way.

In the proposed greedy algorithm, we introduce the unit cost to measure the performance of VMs. The unit cost can be formulated as $\theta_{lij}(t) = \frac{1}{\mu_{ij} + d_{ij}^t(t)} \rho_{lij}$ and $\theta_{lij}(t) = \frac{1}{\mu_{ij} + d_{ij}^d(t)} \rho_{lij}$, where $\theta_{lij}(t)$ and $\theta_{lij}(t)$ are unit costs of a type-$j$ request at $D_l$ served by one class-$i$ VM at $D_l'$ in the reservation and on-demand schemes, respectively. If the request is served at local site (i.e. at $D_l$), the unit costs will be changed to $\theta_{lij} = \frac{1}{\mu_{lj} \lambda^r(t)}$ and $\theta_{lij} = \frac{1}{\mu_{lj} \lambda^{d}(t)}$. The unit cost represents the cost-performance ratio by using one class-$i$ VM instance to serve one type-$j$ application request. In the greedy algorithm, we are seeking to place requests on VMs with the minimum unit cost to achieve the most economic expenditure. The proposed greedy algorithm is presented in Algorithm 1.

Algorithm 1 Greedy Algorithm in Multi-site Cloud

1: for each $D_l \in D$ do
2: Initialize the unit cost set $\Theta = \{\theta_{lij}(t), \theta_{lij}(t), \ldots, \theta_{LMN}, \theta_{LMN}\}$ and set $\Lambda_i(t) = \{\Lambda_i(t), \ldots, \Lambda_i(t)\}$, in which $\Lambda_i(t)$ is $\lambda^i_{init}(t)$ represents the unprocessed type-$j$ application requests.
3: Sort unit cost set $\Theta$ in ascending order.
4: repeat
5: Select the minimum $\theta_{lij}(v = r \ or \ d)$. if $\theta_{lij}$ from reserved VMs (i.e. $v = r$) then
6: Allocate $K_{lij}^{r(t)}$ class-$i$ reserved VMs to serve requests $\Lambda_{lj}^{top}(t) \leq \Lambda_{lj}^{top}(t)$ as long as $\sum_{l'=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} d_{ij}^r(t) \leq \tau_j$ and $K_{lij}^{r(t)} \leq K_{lij}^{init}$. The updated $\Lambda_{lj}^{top}(t) = \Lambda_{lj}^{top}(t) - \lambda^r_{lj}$ and $K_{lij}^{r(t)} = K_{lij}^{init} - K_{lij}^{r(t)}$. If the reserved VMs are not at $D_l$, $\Lambda_{lj}^{top}(t) = \Lambda_{lj}^{top}(t) - \lambda^r_{lj}$.
7: else if $\theta_{lij}$ from on-demand VMs (i.e. $v = d$) then
8: Allocate $K_{lij}^{d(t)}$ class-$i$ on-demand VMs to serve requests $\Lambda_{lj}^{top}(t) \leq \Lambda_{lj}^{top}(t)$ until $\sum_{l'=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} d_{ij}^d(t) \leq \tau_j$ is satisfied. Update $\Lambda_{lj}^{top}(t) = \Lambda_{lj}^{top}(t) - \lambda^d_{lj}$.
9: end if
10: end if
11: until all requests have been processed.
12: end for
13: Calculate the total resource cost $C_{lj}^{tb} = \sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} p_{li} K_{lij}^{r(t)} + \sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{M} d_{ij}^d(t)$
V. EXPERIMENTAL EVALUATION

A. Parameter Settings

We perform simulations to evaluate the proposed optimal resource allocation scheme. Amazon EC2 [7] is one of the most popular clouds allowing application providers to deploy services. To make our evaluation convincing, we employ three geographically distributed Amazon data centers, in which $D_1$ is located at Asia Pacific region Tokyo, $D_2$ is located at EU region Ireland, and $D_3$ is located at US East Virginia. Three classes of VM instances are tested, including small size, large size, and extra large size. The detailed VM instance configuration and price rates can be found from [7]. The application providers supply three multimedia applications. The service rates of different VMs are relative to the number of Amazon EC2 compute units, which can be referred to [8]. The service response time requirements are $\tau = \{0.4s, 0.6s, 0.8s\}$. The arrivals of users’ requests vary with time and the time slot for VM allocation is set as 1 hour.

B. Experimental Results

We first compare the resource cost between the proposed optimal resource allocation scheme, in which the VMs and workload assignment are optimized by solving the resource optimization problem (1), and the proposed greedy algorithm in multi-site cloud. The optimal resource allocation scheme is the global optimal solution but slow to execute, while the greedy algorithm is sub-optimal but lightweight and efficient.

We perform evaluation in a 7-hour period. During the period, the mean request arrival rate is shown in Fig. 2. The corresponding resource cost between the optimal allocation scheme and the greedy algorithm is compared in Fig. 3. From Fig. 3, we can see that the optimal allocation scheme can achieve lower resource cost compared to the greedy algorithm under the same workload. The greedy algorithm only considers the current best choice, while the optimal scheme searches the whole feasible region to reach the global optimal solution. In Fig. 3, the maximal difference of resource cost between the two schemes are 4.2 dollars.

We also evaluate the performance of workload balancing. At the 4th time slot in Fig. 2, data centers 1 and 2 have light workload, while data center 3 reaches a peak of 4000 requests/s. By applying the proposed resource allocation scheme, the workload is balanced as shown in Fig. 4. From Fig. 4, we can find that data centers 1 and 2 process all their local requests and help data center 3 to serve partial workload. In such way, data center 3 can redirect workload to other data centers to avoid local congestion.

Next, we compare the resource cost among the proposed optimal allocation scheme, the greedy algorithm, and the distributed approach in our previous work [4]. The distributed approach in [4] optimizes VM allocation in each data center to minimize the resource cost, but fails to consider the global workload balancing. Fig. 5 shows the comparison result when the request arrival rate increases from 3000 to 7500 requests/s. From Fig. 5, we can find that the proposed optimal resource allocation scheme achieves the lowest resource cost compared to the greedy algorithm and the distributed approach in [4].

VI. CONCLUSION

In this paper, we study the optimal resource allocation problem for multimedia application providers. Based on the Saas architecture and the resource allocation model, we propose the optimal resource allocation scheme in multi-site cloud. In the proposed scheme, we jointly optimize the global workload assignment and the local VM allocation to achieve the minimal resource cost under the service response time requirements. Moreover, we propose a greedy algorithm to efficiently allocate resources in a practical way. The experimental results demonstrate that the proposed optimal resource allocation scheme can effectively achieve the minimal resource cost for multimedia application providers.

REFERENCES