Optimization of Workload Scheduling for Multimedia Cloud Computing

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

***Abstract*— The cloud based multimedia applications have been widely adopted in recent years. Due to the large-scale and time-varying workload, an effective workload scheduling scheme is becoming a challenge faced by multimedia application providers. In this paper, we study the workload scheduling schemes for multimedia cloud. Specifically, we examine and solve the response time minimization problem and the resource cost minimization problem, respectively. Moreover, we propose a greedy algorithm to efficiently schedule workload for practical multimedia cloud. Simulation results demonstrate that the proposed workload scheduling schemes can optimally balance workload to achieve the minimal response time or the minimal resource cost for multimedia application providers.**

I. Introduction

Cloud computing has emerged as a popular computing platform to provide on-demand computation, storage, and communication resources as accessible services for users via the Internet. According to the service provisioning at different levels, three cloud service models have been proposed, namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), among which the SaaS model is the most familiar to individual users. In the SaaS cloud model, application providers rent virtual machines (VMs) from cloud vendors and deliver services to users. In such a way, users send requests for interested applications and receive services from application providers, eliminating the burden of application installation and software maintenance.

Among various applications, cloud-based multimedia applications have been widely used in recent years, like cloud-based photo sharing, on-line video editing, social multimedia applications, etc. For multimedia application providers, there are two important concerns: the *response time* and the *resource cost*. The 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. Due to the delay-sensitive characteristic of multimedia, the response time is taken as an important quality of service (QoS) factor, which is generally contracted into the Service Level Agreement (SLA). Thus, it is necessary for application providers to meet the response time requirement. Besides the response time, the resource cost is another important concern. The resources in cloud include VM, storage, and bandwidth. In this paper, we focus on the computation-orientated multimedia applications, in which the cost on VM rental takes a dominant part of the total resource cost. Generally, two VM pricing schemes are offered by cloud vendors: the *reservation scheme* and the *on-demand scheme*. Price rates in the on-demand scheme are much higher than those in the reservation scheme. However, the application providers have to subscribe a certain number of VMs in advance in the reservation scheme. During service provisioning, if the initially reserved VMs cannot satisfy resource demands, the on-demand VMs can be allocated instantly according to the on-demand pricing scheme. It is significant for application providers to provide satisfactory services under the budget limit.

In cloud platform, multiple classes of VMs are allocated to serve applications. Different classes of VMs generally have different resource capacities in terms of computing units, CPU frequency, memory size, and I/O rate. The computation workload can be balanced by assigning requests to different classes of VMs. However, it is a challenge for multimedia application providers to find the optimal workload scheduling scheme. Firstly, different classes of VMs have different resource capacities and thus process users' requests with different service rates. It is difficult to determine the optimal workload scheduling weight for each class of VMs. Secondly, since the workload in cloud is time varying, the workload scheduling scheme needs to be dynamically optimized to adapt the time-varying workload. Finally, the application providers require an efficient workload scheduling scheme, which can be quickly performed for the practical cloud-based applications.

To address above mentioned challenges, we investigate the optimal workload scheduling schemes in this paper. Specifically, we examine two workload scheduling problems. The first problem is the response time minimization problem which optimizes the workload assignments for the given VMs to minimize the response time. The formulated response time minimization problem is a convex optimization problem. We find the optimal analytical solution for it. The second problem is the resource cost minimization problem which jointly optimizes the workload assignments and the VM allocation. The resource cost minimization problem is a mixed integer non-linear programming, which is known to be NP-hard. Thus, we propose a greedy algorithm to efficiently schedule the workload. The proposed greedy algorithm is a sub-optimal solution, which is demonstrated to perform close to the globally optimal solution in simulations.

II. Related Work

In recent years, we have witnessed a fast development of cloud computing. According to the Gartner Group estimate

978-1-4673-5762-3/13/$31.00 ©2013 IEEE

[1], the worldwide SaaS revenue will reach $14.5 billion in 2012. The major cloud vendors, including Amazon EC2 [2], Microsoft Azure [3], and GoGrid [4] all attract application providers to deploy services on their clouds.

With the development of cloud, some researchers focus on the QoS provisioning in cloud-based multimedia applications [5-7]. Zhu et al. [5] present the concept of multimedia cloud and propose the media edge cloud structure. Nan et al. [6] develop a queuing model to optimize the resource allocation for multimedia cloud. Wu et al. [7] propose a system of utilizing cloud services to support video-on-demand applications. But the authors in [5-7] do not examine the workload scheduling problem. The workload scheduling in the distributed system has always been a challenging research topic. Tai *et al.* [8] propose a burst workload balancer to predict the changes in the user demands and accordingly shift the workload scheduling between the greedy scheme and the random scheme. Silberstein *et al.* [9] present a scheduling algorithm by adapting the multi-level feedback queue approach in a multi-grid environment. However, the proposed workload scheduling schemes in [8], [9] are not optimal and the response time and resource cost are not considered.

III. Workload Scheduling Model

In this section, we present our workload scheduling model for multimedia cloud. Since workload in cloud is time varying, the VMs have to be allocated or released dynamically. Therefore, the time domain is divided into time slots set T and we choose a time slot t ($t \in T$) which is short enough so that the workload in each time slot is constant. Suppose that N classes of VMs are provided for one application. Different classes of VMs have different configurations. A number of VMs in the same class work together as the virtual cluster to provide the faster service.

Fig. 1 shows the workload scheduling model. S is the workload scheduler and $C_1, \ldots, C_N$ are N virtual clusters. Let μ_i be the mean service rate of one class-i VM instance. In time slot t, the numbers of allocated class-i VMs in the reservation scheme and the on-demand scheme are denoted by $K_i^{r(t)}$ and $K_i^{d(t)}$, respectively. Thus, $(K_i^{r(t)} + K_i^{d(t)})$ class-i VMs work together as the class-i virtual cluster with the mean service rate $(K_i^{r(t)} + K_i^{d(t)})\mu_i$, and the service time of a request is assumed to be exponentially distributed with an average of $\frac{1}{(K_i^{r(t)}+K_i^{d(t)})\mu_i}$. According to [6], the arrivals of requests in time slot t can be modeled as a Poisson Process with the mean arrival rate of $\lambda^{(t)}$. As shown in Fig. 1, the incoming requests are distributed to the class-i virtual cluster with the corresponding scheduling weight $\omega_i^{(t)}$ ($\forall i = 1, 2, \ldots, N$). According to the decomposition property of Poisson distribution [10], the arrivals of the scheduled requests to virtual cluster C_i also follow a Poisson Process with the mean arrival rate of $\omega_i^{(t)}\lambda^{(t)}$. Therefore, the service process for the scheduled requests at virtual cluster C_i can be modeled as an $M/M/1$ queueing system [10]. To make the queueing system stable, $\omega_i^{(t)}\lambda^{(t)} < (K_i^{r(t)} + K_i^{d(t)})\mu_i$ is required. The response time T_i at virtual cluster C_i can be formulated as $T_i = \frac{1}{(K_i^{r(t)}+K_i^{d(t)})\mu_i - \omega_i^{(t)}\lambda^{(t)}}$, and thus the mean response time for the application is given by $T = \sum_{i=1}^{N} \omega_i T_i = \sum_{i=1}^{N} \frac{\omega_i^{(t)}}{(K_i^{r(t)}+K_i^{d(t)})\mu_i - \omega_i^{(t)}\lambda^{(t)}}$.

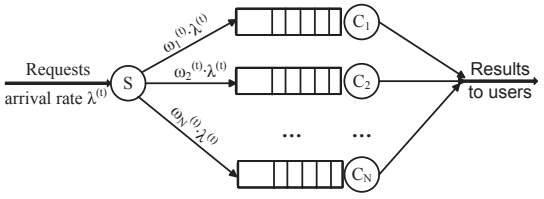

Fig. 1. The workload scheduling model for multimedia cloud.

IV. Response Time Minimization Problem

In this section, we use the proposed workload scheduling model to study the response time minimization problem. In multimedia cloud, different classes of virtual clusters have different service rates. The unbalanced workload scheduling will lead to the episodic local congestion. It is a challenge to optimally schedule system workload to achieve the minimal response time. Therefore, we formulate the response time minimization problem, which can be stated as: to minimize the mean response time by optimizing the workload scheduling weights for different virtual clusters, subject to the queueing stability constraint at each cluster, the workload conserving constraint, and the scheduling weight constraints. Mathematically, the response time minimization problem can be formulated as follows.

$$\begin{aligned} \underset{\{\omega_1,\omega_2,\ldots,\omega_N\}}{\text{Minimize}} \quad & \sum_{i=1}^{N} \frac{\omega_i^{(t)}}{(K_i^{r(t)}+K_i^{d(t)})\mu_i - \omega_i^{(t)}\lambda^{(t)}} \\ \text{subject to} \quad & \\ & \omega_i^{(t)}\lambda^{(t)} < (K_i^{r(t)} + K_i^{d(t)})\mu_i, \\ & \forall i = 1, \ldots, N, \\ & \sum_{i=1}^{N} \omega_i = 1, \\ & \omega_i \geq 0, \ \forall i = 1, \ldots, N. \end{aligned} \tag{1}$$

In the optimization problem (1), the objective function is the mean response time. The constraint $\omega_i^{(t)}\lambda^{(t)} < (K_i^{r(t)} + K_i^{d(t)})\mu_i$ represents the queueing stability constraint at virtual cluster C_i. The constraint $\sum_{i=1}^{N} \omega_i = 1$ guarantees that all workloads should be scheduled for processing. The constraint $\omega_i \geq 0$ represents the scheduling weight constraint.

The response time minimization problem (1) is a convex optimization problem [11]. We use the Lagrange multiplier method [11] to solve the problem and get the optimal analytical solution as follows.

$$\begin{aligned} \omega_i^{(t)} \quad = & \frac{\lambda^{(t)}\sqrt{\xi_i^{(t)}} + \xi_i^{(t)} \sum_{j=1}^{N} \sqrt{\xi_j^{(t)}} - \sqrt{\xi_i^{(t)}} \sum_{j=1}^{N} \xi_j^{(t)}}{\lambda^{(t)} \sum_{j=1}^{N} \sqrt{\xi_j^{(t)}}}, \\ & \forall i = 1, 2, \ldots, N, \end{aligned} \tag{2}$$

where $\xi_i^{(t)} = (K_i^{r(t)} + K_i^{d(t)})\mu_i$ represents the service rate of virtual cluster C_i.

Intuitively, the workload assignment should be proportional to the service rate of virtual cluster C_i. The cluster with the

higher service rate should be scheduled the more requests to process, while the cluster with the lower service rate should be assigned the less workloads. Thus, we proposed a heuristic scheduling scheme, which uses the normalized service rate $\frac{\xi_i^{(t)}}{\sum_{j=1}^{N}\xi_j^{(t)}}$ as the scheduling weight. Compared to the optimal scheduling scheme, the proposed heuristic scheduling scheme is a lightweight but sub-optimal solution.

V. Resource Cost Minimization Problem

In this section, we investigate the resource cost minimization problem. Suppose that application providers initially reserve K_i^{ini} class-i VMs and the deposit is denoted by $\gamma_i K_i^{ini}$. Let the price rates of class-i VM in the reservation scheme and on-demand scheme be P_i^r and P_i^d, respectively. Thus, the total resource cost at time slot t can be given by $\sum_{i=1}^{N}\left(P_i^r K_i^{r(t)} + P_i^d K_i^{d(t)} + \gamma_i K_i^{ini}\right)$. The resource cost minimization problem can be stated as: to minimize the total resource cost by jointly optimizing the workload assignments and the allocated VMs in the reservation and on-demand schemes, subject to the application response time constraint, the queueing stability constraint, the VM reservation constraint, the workload conserving constraint, and the workload scheduling weight constraint. Mathematically, the resource cost minimization problem can be formulated as follows.

$$\begin{aligned}
&\underset{\left\{\substack{K_1^{r(t)},\ldots,K_N^{r(t)},\\ K_1^{d(t)},\ldots,K_N^{d(t)},\\ \omega_1,\ldots,\omega_N}\right\}}{\text{Minimize}} \quad \sum_{i=1}^{N}\left(P_i^r K_i^{r(t)} + P_i^d K_i^{d(t)} + \gamma_i K_i^{ini}\right)\\
&\text{subject to}\\
&\quad \sum_{i=1}^{N}\frac{\omega_i^{(t)}}{(K_i^{r(t)}+K_i^{d(t)})\mu_i-\omega_i^{(t)}\lambda^{(t)}} \le \tau,\\
&\quad \omega_i^{(t)}\lambda^{(t)} < (K_i^{r(t)}+K_i^{d(t)})\mu_i,\\
&\quad \forall i = 1,\ldots,N,\\
&\quad K_i^{r(t)} \le K_i^{ini},\ \forall i = 1,\ldots,N,\\
&\quad \sum_{i=1}^{N}\omega_i = 1,\\
&\quad \omega_i \ge 0,\ \forall i = 1,\ldots,N,
\end{aligned} \tag{3}$$

where τ is the upper bound of response time.

In optimization problem (3), the objective function is the total resource cost. The constraint $\sum_{i=1}^{N}\frac{\omega_i^{(t)}}{(K_i^{r(t)}+K_i^{d(t)})\mu_i-\omega_i^{(t)}\lambda^{(t)}} \le \tau$ represents the response time constraint. The constraint $\omega_i^{(t)}\lambda^{(t)} < (K_i^{r(t)}+K_i^{d(t)})\mu_i$ is the queueing stability constraint. The constraint $K_i^{r(t)} \le K_i^{ini}$ represents that the utilized class-i VMs from the reservation scheme cannot exceed the initially reserved class-i VMs. The constraint $\sum_{i=1}^{N}\omega_i = 1$ is the workload conservation constraint. The constraint $\omega_i \ge 0$ is the workload scheduling weight constraint.

The resource cost minimization problem (3) is a mixed integer non-linear programming, which is known to be NP-hard [12]. The problem can be solved by the branch-and-bound method [12]. However, the application providers require a rapid and efficient scheme, which can quickly adapt to the time-varying workload. Therefore, we propose a greedy algorithm to efficiently schedule workload in a practical way, which is presented in Algorithm 1.

Algorithm 1 Greedy Algorithm for Joint Workload Scheduling and VM Allocation

Input:

The mean request arrival rate $\lambda^{(t)}$; the price rates P_i^r and P_i^d; the number of initially reserved VMs K_i^{ini}; the mean service rate μ_i.

Output:

The scheduling weight $\omega_i^{(t)}$; the required VMs $K_i^{r(t)}$ and $K_i^{d(t)}$; the total resource cost $C^{(t)}$.

Procedure:

1: Compute $q_i^r = \frac{P_i^r}{\mu_i}$ and $q_i^d = \frac{P_i^d}{\mu_i}$, which are the cost rates of using one reserved and on-demand class-i VM to process one unit request, respectively. Let set $Q = \{q_1^r, q_1^d, \ldots, q_N^r, q_N^d\}$.
2: Sort set Q in ascending order.
3: **repeat**
4: Select the smallest q_i^v $(\forall i = 1,2,\ldots,N,\quad v = r \text{ or } d)$ from the set Q
5: **if** q_i^v from the reserved VMs (i.e. $v = r$) **then**
6: Schedule user requests $\lambda_i^{r(t)}$ to the selected class-i reserved VMs as long as the requirements $\frac{1}{K_i^{r(t)}\mu_i-\lambda_i^{r(t)}} \le \tau$ and $K_i^{r(t)} \le K_i^{ini}$ are satisfied. Update $\lambda^{(t)} = \lambda^{(t)} - \lambda_i^{r(t)}$.
7: **else if** q_i^v from the on-demand VMs (i.e. $v = d$) **then**
8: Schedule all unscheduled user requests $\lambda^{(t)}$ as $\lambda_i^{d(t)}$ to the selected class-i on-demand VMs until the requirement $\frac{1}{K_i^{r(t)}\mu_i-\lambda_i^{r(t)}} \le \tau$ is satisfied. Update $\lambda^{(t)} = 0$.
9: **end if**
10: **until** all requests are processed (i.e. $\lambda^{(t)} = 0$).
11: Compute scheduling weight $\omega_i = \frac{\lambda_i^{r(t)}+\lambda_i^{d(t)}}{\lambda^{(t)}}$, and total resource cost $C^{(t)} = \sum_{i=1}^{N}\left(P_i^r K_i^{r(t)} + P_i^d K_i^{d(t)} + \gamma_i K_i^{ini}\right)$.

VI. Simulations

In this section, we perform simulations to evaluate the proposed optimal workload scheduling schemes. Amazon EC2 [2] is the Amazon's cloud computing platform allowing application providers to rent VMs for their services. To make our evaluation convincible, we employ the price rates and VM configurations of Amazon EC2. In our simulations, three classes of VMs are provided. The reservation and on-demand price rates for the three classes of VMs are $P^r = \{0.05\$/\text{h},\ 0.20\$/\text{h},\ 0.40\$/\text{h}\}$ and $P^d = \{0.085\$/\text{h},\ 0.34\$/\text{h},\ 0.68\$/\text{h}\}$, respectively. The numbers of initially reserved VMs are $K^{ini} = \{60, 30, 20\}$, and the service rates for each class of VM instance are $\mu = \{25$ requests/s, 97 requests/s, 185 requests/s$\}$.

We first compare the response time between the proposed optimal scheduling scheme, in which the scheduling weight is

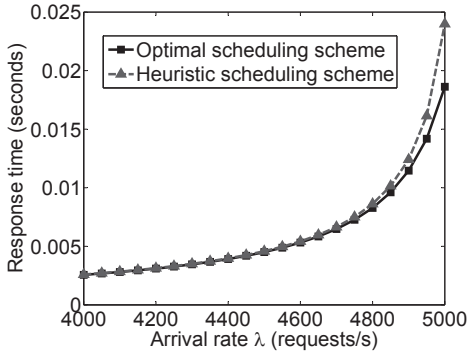

Fig. 2. Comparison of response time between the optimal scheduling scheme and the heuristic scheduling scheme.

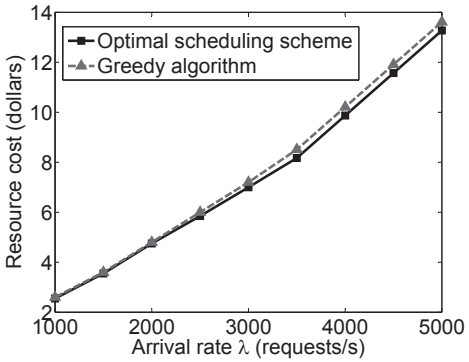

Fig. 3. Comparison of resource cost between the optimal scheduling scheme and the greedy algorithm.

the optimal solution (2), and the proposed heuristic scheduling scheme, in which the scheduling weight is the normalized service rate $\frac{\xi_i}{\sum_{i=1}^{N}\xi_i}$. The optimal scheduling scheme is the globally optimal benchmark, while the heuristic scheduling scheme is a sub-optimal but lightweight solution. The comparison of the response time between the two schemes is shown in Fig. 2. The mean request arrival rate increases from 4000 requests/s to 5000 requests/s. From Fig. 2, we can see that the proposed optimal scheduling scheme achieves the lower response time compared to the heuristic scheduling scheme under the same request arrival rate. Fig. 2 also shows that the heuristic scheduling scheme performs close to the optimal scheduling scheme when the system workload is light. As the request arrival rate increases, the optimal scheduling scheme can optimally change the scheduling weights to adapt the time varying workload, while the heuristic scheduling scheme keeps scheduling weights as constants. Therefore, the difference of the response time between the two schemes increases when the workload becomes heavier. When the arrival rate is 5000 requests/s, the difference of the response time between the two schemes is 0.007 seconds.

Next, we compare the resource cost between the proposed optimal scheduling scheme, in which the scheduling weights and the required VMs are optimally obtained by solving the optimization problem (3), and the proposed greedy algorithm. The optimal scheduling scheme is the globally optimal benchmark but not practical, while the greedy algorithm is sub-optimal but efficient and practical. The comparison of the resource cost between the proposed optimal scheduling scheme and the proposed greedy algorithm is shown in Fig. 3. The mean request arrival rate increases from 1000 requests/s to 5000 requests/s. From Fig. 3, we can see that the optimal scheduling scheme can achieve a lower resource cost compared to the greedy algorithm under the same request arrival rate. Moreover, Fig. 3 shows that the proposed greedy algorithm has a close performance to the optimal benchmark. The greedy algorithm only considers the current best choice but fails to make a global inspection, while the optimal scheduling scheme searches the whole feasible region to find the globally optimal solution. Thus, the greedy algorithm achieves the sub-optimal solution with a lighter computation. The difference of resource cost between the two schemes is 0.34\$ when $\lambda = 5000$ requests/s.

VII. Conclusions

The workload scheduling is a challenge for multimedia cloud computing. To address this challenge, we study the workload scheduling problem in this paper to achieve the minimal response time or the minimal resource cost. Specifically, we first investigate the response time minimization problem by optimizing the workload scheduling weights. The optimal analytical solution is provided for the response time minimization problem. Next, we formulate and solve the resource cost minimization problem by jointly optimizing the scheduling weights and the required VMs. A greedy algorithm is proposed to efficiently solve the resource cost minimization problem. The simulation results demonstrate that the proposed optimal workload scheduling schemes can achieve the minimal response time or the minimal resource cost for multimedia application providers.

References

[1] Gartner estimate. [Online]. Available: http://www.gartner.com/
[2] Amazon ec2. [Online]. Available: http://aws.amazon.com/ec2/
[3] Microsoft azure. [Online]. Available: http://www.windowsazure.com/
[4] Gogrid. [Online]. Available: http://www.gogrid.com/
[5] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *Signal Processing Magazine, IEEE*, vol. 28, no. 3, pp. 59–69, 2011.
[6] X. Nan, Y. He, and L. Guan, "Optimal resource allocation for multimedia cloud based on queuing model," in *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, 2011.
[7] Y. Wu, C. Wu, B. Li, X. Qiu, and F. Lau, "Cloudmedia: When cloud on demand meets video on demand," in *Proc. IEEE Conference on Distributed Computing Systems (ICDCS)*, pp. 268–277, 2011.
[8] J. Tai, J. Zhang, J. Li, W. Meleis, and N. Mi, "Ara: Adaptive resource allocation for cloud computing environments under bursty workloads," in *Proc. IEEE Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, 2011.
[9] M. Silberstein, D. Geiger, A. Schuster, and M. Livny, "Scheduling mixed workloads in multi-grids: the grid execution hierarchy," in *Proc. IEEE Symposium on High Performance Distributed Computing*, pp. 291–302, 2006.
[10] D. Cross and C. Harris, "Fundamentals of queuing theory," *John Wily and Sons*, 1998.
[11] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ Pr, 2004.
[12] J. Karlof, *Integer programming: theory and practice*. CRC Press, 2006.