Bitrate and Task Scheduling in Cloud Computing for Multimedia Big Data

Byeongok Choi
Industrial and System Engineering
KAIST
Taejon, Korea
bo0920@kaist.ac.kr

Chae Y. Lee
Industrial and System Engineering
KAIST
Taejon, Korea
cylee@kaist.ac.kr

***Abstract*— As video traffic increases with plentiful multimedia services and the proliferation of mobile devices such as smartphones, stream mining to extract valuable information out of multimedia big data is garnering attention. By applying cloud computing to stream mining, resource-scarce mobile devices can offload the workloads of heavy applications to a remote cloud. However, resource provisioning for task scheduling is an inherent challenge of stream mining in cloud computing. In this paper we consider problem of resource provisioning and bit rate scaling for multimedia big data processing. We aim to minimize the virtual machine (VM) leasing cost and the classification error cost while satisfying the deadline constraints of workloads which is formulated as a mixed integer nonlinear programming. Deadline based task scheduling and bit rate scaling are developed to find near optimal solution of the NP-hard problem. The upper and lower bounds of the required number of VMs are obtained for infeasible and feasible schedules respectively. Scaling down the highest bit rate first in the bit rate set of a workload is suggested to guarantee the minimum increase of error cost. Our simulation results show the efficiency of bit rate scaling in task scheduling. 5-10% cost reduction is achieved by bit rate scaling in a cloud computing environment.**

Keywords—Stream mining, multimedia big data, cloud computing, task scheduling, bit rate scaling

I. INTRODUCTION

With a huge increase in the amount of visual data and advances in information technology, stream mining has received attention. Stream mining uses multiple multimedia streams relevant to a common scene of interest to extract valuable information for a wide range of applications, such as traffic, wildlife, and weather, out of the streams and then returns it to the users [1]. Some examples are wireless video surveillance, scientific study that uses visual data, visual sensor networks, real-time traffic analysis, and live video streaming. However, stream mining accompanies inherent challenges: First, visual data is characterized by its large-scale. Second, multimedia sources are heterogeneous. Stream mining requires multiple multimedia streams from various mobile device performance to obtain valuable information. Lastly, in the aspect of big multimedia data analysis, mobile devices such as network cameras require high computation performance for data analysis [2].

In stream mining multiple multimedia streams of a common scene of interest are processed and analyzed by a remote cloud in accordance with users' requests. First, users who have the same scene of interest capture videos. For example, in the case of a real-time traffic analysis, network cameras and smartphones from different regions are used to gather visual data. The users then transfer their respective multimedia stream to the cloud. At the same time, they request a workload such as traffic video streaming from other regions. Lastly, the cloud executes workloads from users by utilizing input data and their resources and returns stream mining results to the workload requester.

Most recent studies on cloud resource provisioning problems [3, 4, 5] have focused on cost minimization while satisfying Quality of Service (QoS). Meanwhile, task scheduling is another issue in cloud computing. Assigning tasks to the proper cloud resources according to the size and precedence constraint of tasks has been studied for many years [6, 7, 8]. The relationship of bit rate scaling and classification error also has been studied [1]. Higher video bit rate ensures lower errors in objects classification of the video. Bit rate scaling not only causes error cost variation but also modulates the bit rate of input multimedia data, which affects the task size in task scheduling.

In this paper, we consider the problem of resource provisioning and bit rate scaling for multimedia big data processing with cloud-assisted video stream mining. We aim to minimize the virtual machine (VM) leasing cost and the classification error cost while satisfying the deadline constraint. We develop a bit rate scaling and task scheduling algorithm to enhance efficiency of the cloud resource.

This paper is organized as follows. System model is shown in Section II and a problem description and formulation are presented in Section III. Bit rate scaling and task scheduling algorithm is proposed in Section IV. Performances are evaluated in Section V.

II. SYSTEM MODEL

We consider a stream mining system with users, a workload, and a cloud. Each user gathers a video stream about a scene of interest and transmit it to the cloud. Simultaneously, the user requests the cloud to process a workload by sending input data and workload type information. Input data includes the user's video chunks, its video bit rate, the workload duration and deadline. Workload type is the goal requested by users. Once all workload requests have arrived at the cloud, the cloud is required to schedule tasks of the workloads to appropriate VMs to execute each workload within its deadline.

A. User

There are I users gathering environment information (e.g., monitor a building) about a scene of interest and uploading captured multimedia streams to the remote cloud for a big data

analysis (workload). Possible users include smartphones, network cameras, IP cameras, and cameras in a visual sensor network. User i generates a video stream with multiple default bit rates $r_{i,default}$ according to its device specifications. Note that we assume each user transmits a multimedia stream with a fixed frame per second (fps) and a resolution. Then, only the bit rate affects the video quality, and the bit rate can be adjusted in the cloud to adjust the size of a task for scheduling.

B. Workload

A workload is requested by a user and executed in the cloud by utilizing cloud resources to analyze multimedia streams about a scene of interest. Examples include face recognition, monitoring, transcoding, storage, etc. The workload is composed of a series of tasks and the task processing order is expressed as a workflow. Fig. 1 shows an example of a workload (a surveillance video analysis).

No.	Tasks
1	Background differencing
2	Static object detection
3	Moving object detection
4	Personnel detection
5	Vehicle detection
6	Fusion and output

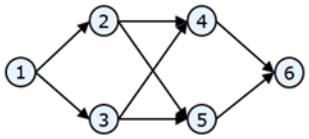

Fig. 1. Tasks set and workflow of the surveillance video analysis application [9]

A workload request is composed of workload type and input data. We denote the workload type k requested by user i by W_i^k and the workload set by $\mathrm{WA} = \{(i,k) \text{ for all } W_i^k\}$. Tasks in W_i^k is given by $t_i^k = \{t_{i,1}^k, \ldots, t_{i,L_k}^k\}$. In the cloud, a multimedia stream is divided into video chunks each of which is a unit raw multimedia stream.

Input data to execute a workload includes workload duration, deadline, video chunks, and video bit rate. Workload duration is determined by the start and end time of requested workload type. It is expressed as *end time – start time* = $T_{i,end}^k - T_{i,start}^k$. Deadline $T_{i,deadline}^k$ is the requested due time of the workload. All available video chunks necessary to process W_i^k are supported and surveyed by users $j \in vc(W_i^k)$. The bit rate of input video chunks has values $r_{j,b} \in R_i^k$ for all W_i^k. Finally, the input data includes the task precedence relationship.

C. Cloud

We consider the cloud as IaaS (Infrastructure as a Service) with a single type of VM. We only consider CPU clock frequency f_c $(cycles/time)$ as a unit resource of the VM. In Amazon, the prices (tariffs) of the reservation plan are lower than those of the on-demand plan (i.e., time discount rates are only offered to the reserved and prepaid resources). Cloud resources are reserved and reservation cost C is proportional to the number V of leased VMs. We assume that users transmit video streams and these streams are encoded and integrated in the cloud for each requested workload. Bit rate scaling is then executed in the integration step and the size of the cloud is adjusted before the scheduling.

III. Problem Description and Formulation

A. Processing Time of Task

There is no simple analytic expression of task processing time. In this study task execution time is estimated by assuming that the task processing time is an increasing function of the input video chunk size and the complexity of task processing. Each task $t_{i,l}^k$ requires specific CPU cycles to execute a unit input video chunk size. Usually, in video streaming, the length of a video chunk is set to two seconds and its size is proportional to the bit rate: $S(r_{j,b}) = 2 \cdot r_{j,b}$. Two seconds would then be a time unit for the stream mining.

Note that cloud resource requirement for a workload is proportional to the transmission rate [1]. Higher transmission rate requires longer service time. In this paper, for workload W_i^k, video chunks of related users $j \in vc(W_i^k)$ are necessary and the size of video chunks $I_j(W_i^k)$ by user j for W_i^k is proportional to its workload duration:

$$I_j(W_i^k) = S(r_{j,b}) \cdot (T_{j,end}^k - T_{j,start}^k)/2 \quad (1)$$

Thus, the total size of input video chunks for workload W_i^k is denoted by $I_i^k = \sum_{j \in vc(W_i^k)} I_j(W_i^k)$. The complexity of task processing also affects the processing time [21]. As the complexity $L_i{}^k$ of task processing increases, the task processing time will also increase. We denote the processing time of task l of W_i^k by

$$TP(t_{i,l}^k) = I_i^k L_i^k / f_c \quad (2)$$

B. Processing Time of Workload

The processing time of a workload depends on the processing time of its tasks and the task scheduling strategy. We define a task assignment function $A(t_{i,l}^k, V) = (t_{i,l_{start}}^k, v)$ that returns the task start time and the assigned VM index. The start time of task j of W_i^k is determined by the task precedence relationship as follows:

$$t_{i,j_{start}}^k \geq t_{i,l_{start}}^k + TP(t_{i,l}^k) \text{ for all } l \quad (3)$$

In (3) task l precedes task j. The processing time of workload W_i^k is then the sum of the start time of its last task and its processing time:

$$WP(W_i^k) = t_{i,L_{start}}^k + TP(t_{i,L}^k) \quad (4)$$

C. Problem Formulation

As mentioned in Section II *B*, the bit rate of input video chunks has values in the set R_i^k. We define the bit rate distribution of all workloads as $\mathrm{R} = \{R_i^k \text{ for all } (i,k) \in \mathrm{WA}\}$.

1) Error Cost: Clearly, users could acquire better classification performance with video chunks of higher bit rate. In [1], authors define the classification error cost of a user as a decreasing function of the input video bit rate. We consider the sum of the error cost per workload with the error cost $c_j(r_{j,b})$ proposed in [1].

$$\sum_{(i,k) \in WA} \sum_{j \in vc(\mathrm{W}_i^k)} c_j(r_{j,b}) \quad (5)$$

This approach gives a weight to error cost of each user whose stream is used more for the workload.

2) Leasing Cost: Leasing cost is proportional to the number of VMs leased. Thus the total leasing cost becomes $C \cdot V$, where C is the cost of a unit VM.

3) Deadline Constraint: As a user has the deadline of the requested workload k, the workload processing time should be smaller than the deadline:

$$WP\left(W_i^k\right) \le T_{i,deadline}^k \quad (6)$$

Therefore, to minimize the total cost while satisfying deadline constraint, we propose an algorithm that solves the following problem

$$\min_{R,A(t_{i,l}^k,V)} \sum_{(i,k)\in WA} \sum_{j\in vc(\mathrm{W}_i^k)} c_j(r_{j,b}) + \alpha \cdot C \cdot V \quad (7)$$

$$\text{s.t.} \quad t_{i,j_{start}}^k \ge t_{i,l_{start}}^k + TP\left(t_{i,l}^k\right) \text{ for all } i,k \text{ and } l \quad (8)$$

$$WP\left(W_i^k\right) \le T_{i,deadline}^k \quad (9)$$

$$r_{j,b} \in R_i^k \text{ for all } \mathrm{W}_i^k,\ V \ge 0,\ integer$$

Equation (7) is the sum of the error cost and the leasing cost, α is a weight to the leasing cost. (8) is the task precedence constraint, and (9) is the deadline constraint. The proposed minimization problem is a mixed integer nonlinear programming model which is known as an NP-hard problem. Since the task scheduling in cloud resource is difficult to get the optimal solution due to its combinational nature and potential existence of multiple local minima in the search space, we propose a heuristic procedure.

IV. Proposed Algorithm

This section introduces the bit rate and task scheduling algorithm for multimedia big data in mobile cloud. The goal is to find the optimal number of VMs and the optimal bit rate distribution. It includes three parts: 1) initial VM estimation; 2) task scheduling with deadline constraint; and 3) VM and bit rate scaling.

Initially, we estimate the initial number of VMs, *V_initial*, required to execute workloads. Given *V_inital*, tasks arriving at the cloud are scheduled by the deadline-based scheduling algorithm discussed in Section IV *B*. Two possible cases for the scheduling results are: 1) there are workloads that do not satisfy the deadline constraints, 2) all workloads satisfy the deadline constraints. According to the scheduling results, the number of VMs and the bit rate distribution of input data are adjusted to find the optimal *V* and *R* that minimize the machine leasing and error cost while satisfying the deadline constraints.

A. Initial VM Estimation

We set the initial number of VMs for task scheduling in such a way that there are cloud resources to meet the minimum requirement for the workload execution during the time period of VM leasing. First, we place all tasks sequentially, then the processing time of the requested workloads is

$$\sum_{(i,k)\in WA} \sum_{l\in\{1,2,\ldots,L_k\}} TP\left(t_{i,l}^k\right) \quad (10)$$

Given the time period *T* of VM leasing, the amount of available resource of one VM becomes $T \cdot f_c$. The estimated number of VMs is then the upper bound of the processing time of the workloads requested divided by the available resources of one VM as in (11).

$$V_{initial} = \left\lceil \sum_{(i,k)\in WA} \sum_{l\in\{1,2,\ldots,L_k\}} TP\left(t_{i,l}^k\right) / (T \cdot f_c) \right\rceil \quad (11)$$

B. Task Scheduling with Deadline Constraint

Our system model has three characteristics: associated tasks, multiple workloads, and heterogeneous VMs. Since the workload scheduling in this paper is for associated tasks, we use dynamic provisioning dynamic scheduling (DPDS) [10], which is an online algorithm that provisions resources within their budget and schedules tasks at runtime. It consists of two main parts: a provisioning procedure and a scheduling procedure. DPDS guarantees that tasks from lower priority are always deferred when higher-priority tasks are requested, but lower-priority tasks can still occupy idle VMs when all higher-priority tasks are scheduled. We set the priority of the workloads according to their deadlines. A workload with earlier deadline has higher priority than other workloads. This guarantees that tasks with early deadline will be scheduled with a fixed number of VMs. Algorithm 1 shows the 'deadline-based scheduling algorithm'.

Algorithm 1 Deadline-based task scheduling

$\mathrm{S}_{\mathrm{arrival}} \leftarrow$ empty arrival set
$\mathrm{S}_{\mathrm{current}} \leftarrow$ empty current set
$\mathrm{VM} \leftarrow$ vm list set
$\mathrm{VM}_{\mathrm{time}} \leftarrow$ empty vm time set
$\mathrm{T}_{\mathrm{assigned}} \leftarrow$ empty assigned task set
for task l in workload k requested by user i **do**
  $\mathrm{INSERT}(\mathrm{t}_{i,l}^{\mathrm{k}}, \mathrm{S}_{\mathrm{arrival}})$
end for
sort $\mathrm{S}_{\mathrm{arrival}}$ in early deadline order
while $\mathrm{S}_{\mathrm{arrived}} \neq 0$ **do**
  for task $\mathrm{t}_{\mathrm{i,l}}^{\mathrm{k}}$ in $\mathrm{S}_{\mathrm{arrival}}$ **do**
    $\mathrm{S}_{\mathrm{current}} \leftarrow \mathrm{t}_{\mathrm{i,l}}^{\mathrm{k}}$ if $\mathrm{TP}^{\mathrm{k}}\left(t_{i,l}^k\right) \in \mathrm{T}_{\mathrm{assigned}}$
  end for
  for $\mathrm{t}_{i,l}^{\mathrm{k}} \in \mathrm{S}_{\mathrm{current}}$ **do**
    $\mathrm{T}_{assigned} \leftarrow \mathrm{t}_{i,l}^{\mathrm{k}}$
    $\mathrm{v} \leftarrow \mathrm{MIN}(\mathrm{VM}_{\mathrm{time}})$
    $\mathrm{INSERT}(\mathrm{t}_{i,l}^{\mathrm{k}}, \mathrm{VM(v)})$
    $\mathrm{INSERT}(TP\left(t_{i,l}^k\right), \mathrm{VM}_{\mathrm{time}})$
    Remove $\mathrm{t}_{\mathrm{i,l}}^{\mathrm{k}}$ from $\mathrm{S}_{\mathrm{current}}$
  end for
while for

C. Tradeoff between VM and Bit Rate Scaling

In task scheduling, there are trade-offs between VM scaling and bit rate scaling. VM scaling affects available cloud resource and the leasing cost. When V is decreased to V', the available resource is decreased by $(V - V') \cdot f_c$. and the leasing cost is decreased by $\alpha \cdot (V - V') \cdot C$. On the contrary, when V is increased to V', the leasing cost is increased by $\alpha \cdot (V' - V) \cdot C$. Note that the bit rate scaling affects the processing time of a task and the error cost. When the bit rate of a user *j*, $r_{j,b}$ for the workload W_i^k is reduced to $r_{j,b'}$, the input data size of user *j* is reduced by $S\left(r_{i,b}\right) - S(r_{i,b'}) \cdot (T_{i,end}^k - T_{i,start}^k)/2$. Thus, the total size of input data for workload $\mathrm{W}_i^{\mathrm{k}}$ is decreased, and the processing time of workload $\mathrm{W}_i^{\mathrm{k}}$ is reduced by $\sum_{j\in vc(W_i^k)} (S\left(r_{i,b}\right) - S(r_{i,b'})) \cdot (T_{i,end}^k - T_{i,start}^k) L_l^k / 2 f_c$. On the other hand, the error cost is increased by $\mathrm{c_j}(r_{j,b'}) - c_j(r_{j,b})$. In the case of scaling the bit rate up, the processing time of a task is increased but the error cost is decreased. Given the number of VMs V, a local optimal point (V, R^*) is defined as the point

that satisfies the deadline constraint with the highest bit rate after task scheduling. Fig. 2 illustrates the relationship between the number of VMs and the bit rate scaling for the deadline constraint and the local optimal points. Fig. 2 (a) shows that if the current point (V, R) satisfies deadline constraint, then all points (V, R') whose bit rate $R' < R$ satisfy the deadline constraint. Fig. 2 (b) shows that if the current point satisfies the deadline constraint, the bit rate of the local optimal point is higher than or equal to that of the current point. On the other hand, Fig. 2 (c) shows that if the current point does not satisfy the deadline constraint, the bit rate of the local optimal point will be lower than or equal to that of the current point. Lastly, Fig. 2 (d) shows that as the number of VMs increases, the bit rate of the local optimal point increases.

Given multiple local optimal points, we are interested in the optimum point (V^*, R^*) which ensures the lowest total cost.

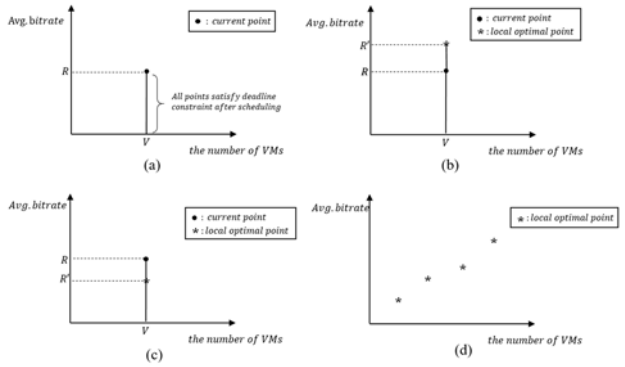

Fig. 2. Illustration of VM and bit rate scaling and local optimal points

In the VM and bit rate scaling algorithm, VMs are increased or decreased between the upper and lower bounds. When a solution does not satisfy the deadline constraint, the bit rates are scaled down. The number of VMs is increased as far as the increase of leasing cost is smaller than the reduction of error cost. Otherwise, it is decreased and bit rates are adjusted. The algorithm stops when the decrease of VM leasing cost is smaller than the increase of error cost.

V. Performance Evaluation

In this chapter, we present the results of simulations to evaluate the performance of our proposed bit rate and task scheduling algorithm. The overall system is generated by Matlab and the algorithm is simulated in that system. We conduct three sets of simulations: 1) scheduling results 2) maximum number of VMs for scaling; and 3) task scheduling with and without scaling.

A. Simulation Setup

We generate three different types of workloads [11]: a parallel model; a sequential model; and a complicated model. In the workload, each task has different task complexity for execution. We randomly assign complexity according to the workload types. The size of each task is proportional to the total size of input data multiplied by the complexity. Ten users who are interested in the same scene request ten workloads. Each user has a specific default bit rate of video chunks according to their device. We refer to the recommended video bit rate for standard dynamic range uploads for YouTube [11]. Table I presents available default bit rates for users. We assume that bit rates have a discrete uniform distribution of $U(r_{default} - 2, r_{default} + 2)$, where $r_{default} \in \{2.5, 5, 6, 16, 35\}$. We set the resource capacity of a VM to 10 Mbps and the lease time to 60 seconds. In addition, we set the error cost as $c_j(r_{j,b}) = 2^{-r_{j,b}}$ for $j = 1,2,\dots,10$ and $r_{j,b} \in r_{default}$. Complexity of the workload is assigned according to its types: 1) a complicated model: 1.1; 2) a sequential model: 0.9; 3) a parallel model: 1.0. We assume that each user requires different video chunks of users. User 1, 4, and 10 require video chunks of all users and other users require video chunks of random users. Lastly, each workload requires different length of video chunks for the execution.

TABLE I. Default bit rates and bit rate range for scaling

Default bit rate	Bit rate range
35 Mbps	[33, 34, 35, 36, 37]
16 Mbps	[14, 15, 16, 17, 18]
8 Mbps	[6, 7, 8, 9, 10]
5 Mbps	[3, 4, 5, 6, 7]
2.5 Mbps	[0.5, 1.5, 2.5, 3.5, 4.5]

B. Main Results

Table II shows the VM estimation with $V_{initial} = 17$ when we set the VM cost as \$10 per minute with $\alpha = 0.07$. Then, we conduct task scheduling with Algorithm 1. First, there are four tardy workloads (Workload 1, 2, 4, and 7). Bit rate is scaled down to each tardy workload to decrease the video chunk size to satisfy the deadline constraint. After bit rate scaling, the average bit rate of video chunks decreases by 1.2459 Mbps and then the error cost increases by 3.8728. As a result, the total cost increases by 1.7698.

TABLE II. VM ESTIMATION

Estimation	
Total workload size	10051.5 MB
Lease time	60 sec
Estimation	16.7525
Initial number of VMs	17

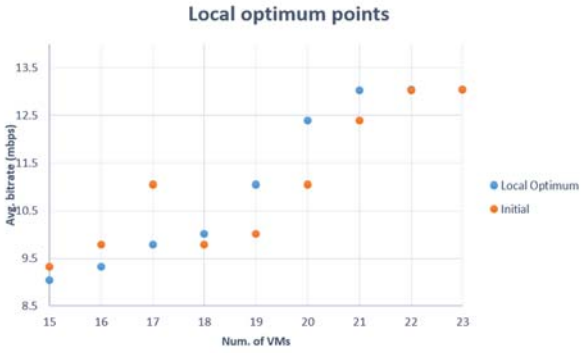

Fig. 3. Local optimum points

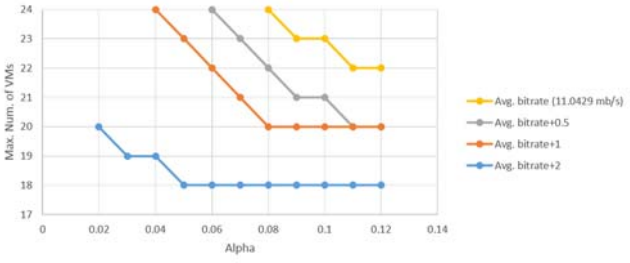

Figure 4. Maximum number of VMs for scaling

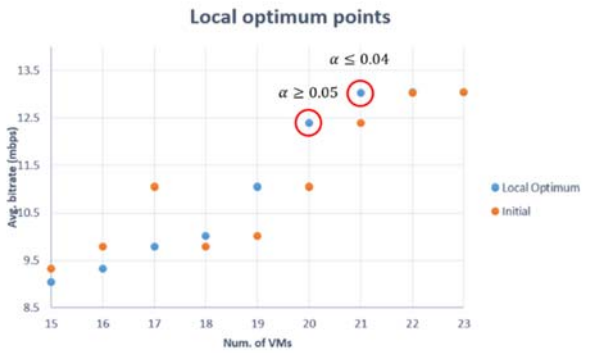

Fig. 5. Alpha and the optimal points

Next, we investigate the range of VM scaling. Using (12), we have $V_{up} = 23$ and $V_{low} = 15$. The overall scheduling results are shown in Fig. 3. The scaled bit rate of 9.7970 of $\mathrm{VM}_{\mathrm{initial}} = 17$ would be the initial bit rate for $\mathrm{VM} = 18$. Due to the increased number of VMs the bit rate for the new local optimum is increased to 10.0282. On the contrary, the scaled bit rate of 9.797 of $\mathrm{VM}_{\mathrm{initial}} = 17$ becomes the initial bit rate for $\mathrm{VM} = 16$. Since the available resource has decreased, the bit rate for the new local optimum is decreased to 9.3421. Finally, we compare the local optimums and select the one that has the minimum total cost. In our simulation, $P(V, R_{avg}) = (20, 12.3855)$ is selected as the optimum point as shown in Fig. 5.

Fig. 4 shows the effect of alpha and the average bit rates on the maximum number of VMs for scaling. In (12), the bit rate distribution and the value of alpha are the main factors that determine the value of V_{incre}. The figure shows that given a value of alpha, as the average bit rate increases, the maximum number of VMs for scaling $\mathrm{V}_{\mathrm{incre}}$ decreases. Thus, as users have higher bit rate in the system, the search space for VM, $[V_{low}, V_{up}]$, decreases more. Second, given the average bit rate, as alpha increases, the maximum number of VMs for scaling $\mathrm{V}_{\mathrm{incre}}$ decreases. Therefore, when we emphasize the importance of the leasing cost, the search space for VM, $[V_{decre}, V_{incre}]$, decreases.

Fig. 5 graphically plots the results of our simulation depending on the value of alpha. As alpha increases, the marginal leasing cost of a VM increases. Thus, among local optimum points, the point with a smaller number of VMs is selected as the optimum. On the contrary, as alpha decreases, the marginal leasing cost of a VM decreases. Therefore, the local optimum point with a larger number of VMs is selected as the optimum.

VI. CONCLUSION

This paper considers task scheduling requested by mobile multimedia users. Cloud computing resource is assigned to execute and meet the requested workloads. Virtual machines are leased and adjusted to schedule the workloads such that the machine leasing cost and processing error cost are minimized. Scheduling algorithms are provided to decide the optimal bit rate distribution of users input data and the number of VMs to lease for a multimedia big data analysis in mobile clouds. In the proposed scheduling algorithm, we consider the deadline of each workload requested by users in stream mining.

Initial number of virtual machines are computed by dividing the total processing time of tasks with cloud lease time. Tasks in a workload are assumed processed sequentially. Deadline based task scheduling is performed by considering parallel machine processing. If any workload does not satisfy its deadline, bit rate scaling is performed starting from the highest bit rate workload. If a local minimum schedule is obtained, the number of virtual machines are increased or decreased to have better solutions. It is increased as far as the reduction of error cost due to the increased bit rate is larger than the increase of leasing cost. On the other hand, it is decreased as far as the reduction of leasing cost is larger than the increase of error cost.

Computational results show that the bit rate scaling is effective for the task scheduling in mobile clouds. It reduces the total cost 5 ~ 10 % compared to the scheduling without scaling.

REFERENCES

[1] S. Ren and M. Schaar, "Efficient resource provisioning and rate selection for stream mining in a community cloud," IEEE Transactions on Multimedia, vol. 15, pp. 723-734, 2013.

[2] A. S. Kaseb, A. Mohan, and Y. H. Lu, "Cloud resource management for image and video analysis of big data from network cameras," International Conference on Cloud Computing and Big Data, pp. 287-294, 2015.

[3] F. Chen, C. Zhang, F. Wang, J. Liu, X. Wang, and Y. Liu, "Cloud-Assisted Live Streaming for Crowdsourced Multimedia Content. IEEE Transactions on Multimedia," vol. 17, pp. 1471-1483, 2015.

[4] Q. He, J. Liu, C. Wang, and B. Li, "Coping with Heterogeneous Video Contributors and Viewers in Crowdsourced Live Streaming: A Cloud-Based Approach," IEEE Transactions on Multimedia, vol. 8, pp. 916-928, 2016.

[5] M. M. Hassan, B. Song, M. S. Hossain, and A. Alamri, "QoS-aware resource provisioning for big data processing in cloud computing environment," International Conference on Computational Science and Computational Intelligence, vol. 2, pp. 107-112, 2014.

[6] M. Rodriguezand and R. Buyya, "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds," IEEE Transactions on Cloud Computing, vol. 2, pp. 222-235, 2014.

[7] C. W. Tsai and J. J. Rodrigues, "Metaheuristic scheduling for cloud: A survey," IEEE Systems Journal, vol.8, pp. 279-291, 2014.

[8] V. Arabnejad, K. Bubendorfer, B. Ng, and K. Chard, "A Deadline Constrained Critical Path Heuristic for Cost-Effectively Scheduling Workflows," IEEE/ACM 8th International Conference on Utility and Cloud Computing, pp. 242-250, 2015.

[9] M. Ebrahimi, A. Mohan, S. Lu, and R. Reynolds, (2015, October). "TPS: A task placement strategy for big data workflows," IEEE International Conference on Big Data, pp. 523-530, 2015.

[10] M. Malawski, G. Juve, E. Deelman, and J. Nabrzyski, "Cost-and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds," Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, pp. 1-11, 2012.

[11] P. Zhong and C. H. Chi, "Streaming Data Rate Prediction Model for Multimedia Service Workflow," IEEE International Conference on Service Systems and Service Management, pp. 1-6, 2007.