International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 9 Issue 3 –SEPTEMBER 2014.

Efficient Job-Dispatching Algorithm in a Multimedia Cloud Service

C.N.Vanitha[1] and G.Hemaladevi[2]

Assistant Professor[1], Department of Computer Science and Engineering, Mahendra College of Engineering, Salem

[2]PG Scholar, Department of Computer Science and Engineering, Mahendra College of Engineering, Salem

rushtovanitha@gmail.com[1],
hdevi41@gmail.com[2]

Abstract-The cloud readiness segment of this study offers a regional view of the requirements for broadband and mobile networks to deliver next-generation cloud services. The enhancements and reliability of these networks will support and increased adoption of business consumer cloud computing solutions that deliver basic as well as advanced application services. For example, consumers expect to be able to communicate with friends as well as stream music and videos any time, any place. Business users require continuous access to business communications and mobile solutions for videoconferencing and mission-critical customer and operational management systems. Download and upload speeds as well as latencies are essential measures to assess network capabilities for cloud readiness. Adopting the framework of Lyapunov optimization, we propose the control algorithm REQUEST to dispatch transcoding jobs. We characterize the energy–delay tradeoff of the REQUEST algorithm numerically and derive the performance bounds theoretically. The REQUEST algorithm is robust to inaccuracy of the transcoding time estimation. We compare the performance of the REQUEST algorithm with Round Robin and Random Rate algorithms using simulation and real trace data. The proposed request algorithm can be applied in cloud-assisted multimedia transcoding service. we propose an online algorithm that dispatches the transcoding jobs to service engines reduce energy consumption achieving the queue stability.

Keywords—Energy efficiency, job dispatching, transcoding as a service (TaaS)

I. INTRODUCTION

In the popular of mobile devices, users have to increasing the usage of online video consumption on devices. According to a Cisco VNI report [1], for global Internet video traffic will contribute 79% of all Internet traffic in 2017: up from 65% in 2012. This trend of people video will be used for much process; however, they will be hampered by the limited bandwidth and inherent nature of stochastic

Wireless channels (e.g., without sound and shadowing effects), it can humiliate the user's experience to watching the videos. Transcoding technology is introduced to become accustomed the videos according to available bandwidth in the purpose of user requirement.

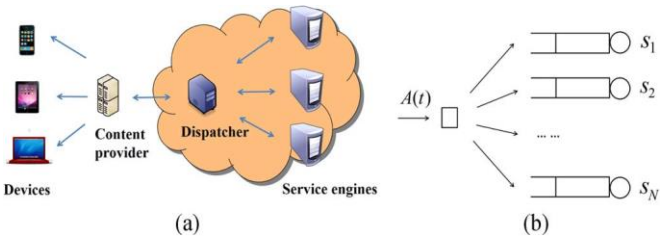


Fig.1.Overview of a cloud-assisted multimedia transcoding platform

The content provider can send transcoding jobs to the cloud. A dispatcher at the front end of the cloud receives transcoding jobs and dispatches them to a set of service engines at the back end for transcoding. (a) System architecture.(b) System model.

The main process of a content provider is to transcode the same video into multiple rates or multiple formats for users' need. In adding together, the resolution size of a video can be reduced because of users can watch the video smoothly over the network. On the other hand, such a transcoding process is computation demanding for the content provider. It is a challenge for the content provider to maintain the low delay for transcoding when many requests arrive. Large scale platform also support transcoding process.

Our contributions in this paper are multirole.

1) Adopting the framework of Lyapunov optimization, we propose the control algorithm REQUEST to dispatch transcoding jobs. We characterize the energy–delay trade-off of the REQUEST algorithm numerically and derive the performance bounds theoretically.

2) We are learning about the robustness of the REQUEST algorithm. Numerical results show that, given the

incorrectness of estimating the transcoding time, and the time average energy consumption and queue backlog error is small. Therefore, the REQUEST algorithm is robust to inaccuracy of the transcoding time estimation. Next process is to be comparing the performance of the REQUEST algorithm with Round Robin and Random Rate algorithms using the simulation of real data. The results will show appropriately choosing the control variable, the REQUEST algorithm outperforms the other two algorithms, with smaller time average energy consumption were as achieving queue stability.

II. RELATED WORK

Previous works have to investigate the transcoding in distributed systems. In this tasks are planned for a cluster-based web server to minimize the total processing time by predicting the processing time per individual task. In the transcoding time is estimated, and an estimation model for load distribution among distributed servers is imported. These two works did not examine the robustness of the scheduling algorithms for the case that the estimation model is not accurate. Our planned algorithm is robust to the inaccuracy of the estimated time.

Now Cloud computing process is to increase the performance of transcoding.Cloud computing applications are delivered as the services over the internet [6]. Hadoop-based cloud for transcoding media content is absorbed; it can greatly improve encoding times. In [2], a cloud transcoder to bridge the gap between videos and mobile devices is proposed, reducing the transcoding burden on mobile devices. In simulation is provided for a cloud transcoding system with cache capability and the proper cache sizes and the number of computers are explored to operate effectively in the cloud. In [3], a load-sharing algorithm in a transcoding cluster is provided, and in [4], a scalable distributed media transcoding system that can reduce the transcoding time is presented. In queue waiting time of transcoding servers is used to make an admission control for video streams and job dispatching for video transcoding to prevent jitters. In virtual machine provision for video transcoding, is considered as the cost-efficient. In [5], mechanisms for allocation and reallocation of virtual machines and video transcoding servers are provided.

III. SYSTEM MODELS AND PROBLEM FORMULATION

A. Arrival Model

First we think a discrete time slot model. The length of a time slot is τ. assume that τ is small such that there is most one transcoding job arriving to the dispatcher for each time slot. We denote p as the probability of one arrival to the dispatcher for each time slot and $1 - p$ if there are no arrivals.

We assume that the transcoding time is needed for arriving a job at each time slot is associated with CPU speed of the service engine. Suppose we have N service engines for transcoding. Each service engine can operated in different CPU speed s_i, where $i = 1, 2. . . N$. Without loss of generality, we assume that $s_1 \leq s_2 \leq \cdot \cdot \cdot \leq s_N$. The service engine in faster CPU speed can have less completion time for transcoding. We note $A(t)$ as the transcoding time needed for the arrival at time slot t by a baseline server, which has a CPU speed S. Assume that by statistical learning, the dispatcher can estimate the transcoding time of each job, i.e., $A(t)$. Then, if the transcoding job is dispatched to the ith service engine, the transcoding time at the ith service engine is $A_i\ (t) = SA\ (t)/s_i$. The transcoding time of the same arrival can be different for service engines, and thus, the dispatcher needs to decide which service engine should process the arriving job.

B. Queuing Model

The service engines as a set of queues, as shown in Fig. 1(b). To describe the dynamics of these queues, define queue length Q (t) as the unfinished transcoding time of jobs in each service engine at time slot t, i.e., Q (t) = $\{Q_1\ (t), Q_2\ (t). . . Q_N\ (t)\}$. The queue of the ith service engine evolves according to

$$\boldsymbol{Q_i\ (t+1) = \max\ [Q_i\ (t) - \tau, 0] + A_i\ (t)\ 1_{\{u\ (t)\ =i\}}}$$

Where $A_i(t)$ is the transcoding time of an arrival at time slot t for the ith service engine, and 1 is an indicator function that is 1 if $u(t) = i$ and 0 otherwise. If $u(t) = i$, the arrival is dispatched to the ith service engine, and the queue length is increased by $A_i(t)$; otherwise, no arrival occurs to the ith service engine. And also we observe the queue length of service engines for each time slot.

C. Energy Consumption Model

Each service engine like as a physical machine. Particularly, consider the working about energy consumption in the service engine that is a leading term for the energy consumption in the distributed servers[7].As such, we overlook for other sources of energy consumption in the service engine, e.g., memory and network. Assume that each service engine operates in a constant CPU speed when processing transcoding jobs takes. Its resulted energy consumption is assumed to be a function of CPU speed. Then dispatcher dispatches the transcoding job to the ith service.

D. Problem Formulation

Initially, the dispatcher routes the transcoding job to the

service engine with the least queue backlog, it can reduce the delay of the transcoding job; still, it would incur large energy consumption if many transcoding jobs are dispatched to service engines with fast CPU speed. The engine at time slot *t*, the energy consumption on the *i*th service engine is $A_i(t)\ \kappa s\alpha i$, where $A_i(t)$ is the transcoding time for the *i*th service engine, and $\kappa s^{\alpha}{}_i$ is the power that is a convex function of CPU speed. Normally, α is set to be process. In adding, without loss of generality, set the constant parameter $\kappa = 1$. If the job is dispatched to the service engine with fast CPU speed, it produces result in high energy consumption. If the service engine has no transcoding jobs to process, the service engine can be set to sleep mode, resulting in very small energy consumption that can be negligible.

In addition, we ignore the resulted energy and time due to a transition from sleep mode to running mode of a service engine. For the computation-intensive transcoding jobs, the computation energy and time are the first-order component for the total energy and time, whereas the energy and time due to the transition overhead from the sleep mode to running mode are the second-order component. Although the overhead is also critical, its effect could be ignored for the decision of job dispatching.

IV. ONLINE DISPATCHING ALGORITHM

Here, we agree to the Lyapunov optimization framework to solving the problem of the optimization and design an process of online dispatching algorithm.

A. Algorithm Design under I.I.D. Transcoding Time

Let we assume that the transcoding time of an arriving job by the baseline server $A(t)$ is independent and identically distributed (i.i.d.) for every time slot. Then, $Ai\ (t)$ is also i.i.d. for the *i*th service engine. We define the quadratic Lyapunov function

$$L\ (Q\ (t)) = \frac{1}{2}\sum_{i=1}^{n} Q_i(t)^2$$

Algorithm 1 REQUEST Algorithm

Input: Q (*t*)

Output: *u* (*t*)

1. At the starting off each time slot is *t*, observe the queue length Q (*t*).
2. Determine *u* (*t*) that minimizes the bound of the *drift-plus-penalty* function.
3. Update the queue Q(*t*).

B. Performance Analysis

Theorem 1: We assume that the arrival of transcoding jobs is strictly within the capacity region. Capacity region is defined as the set Λ of all non negative $a_i(t) = A_i(t)1_{\{u(t)=i\}}$, for which

$$E\{a_i(t)\} < T \qquad \forall i$$

Also assume that $E\{L\ (Q\ (0))\} < \infty$. Then for any control variable $V > 0$, the online algorithm can stabilize the system, then the resulted time average energy consumption and queue backlog satisfying the following equation:

$$\bar{E} \le E^* + \frac{B}{V}$$

$$\bar{Q} \le \frac{B + V E}{\epsilon}$$

Where ε is a constant and $E*$ is a theoretical lower bound on the time average energy consumption

C. Robustness Analysis

In the previous, we can study how to observe the queue length accurately at the beginning of each time slot before making the decision of dispatching transcoding jobs. However, this observation may not be accurate. we now the robustness of the REQUEST algorithm under the inaccurate queue length information.

V. NUMERICAL CHARACTERIZATION OF THE REQUEST ALGORITHM

Here, first we construct a statistical model to estimate the transcoding time. Subsequent that, we describe the tradeoff between the energy consumption and queue backlog for the REQUEST algorithm in this paper, then, we study the robustness of the REQUEST algorithm given that the estimated transcoding time is not accurate.

A. Statistical Model of the Transcoding Time

The transcoding time as a function for the model of the file size of a video, which is given by the variables of transcoding time

$$A = FR$$

Where *F* is the file size, and *R* is a random variable that denotes the transcoding time for a unit of file size.

Particularly, X reflects the difficulty of the transcoding process, which is determined by the conversion of resolution size, bit rate and frame rate, etc. In this paper, this paper we only consider the conversion of resolution size; other transcoding parameters linger as our future work.

B. Settings for Numerical Characterization

Table I summarizes the parameter settings. We set p = 0.8.The CPU speed of a baseline server is S = 3.2 GHz. assume that there are ten service engines, the CPU speeds of which range from 2.0 to 2.9 GHz increased by 0.1 GHz. We also set κ = 1 and α = 3 for the energy model. Each service engine is assumed that have an empty queue at the first time slot.

Table 1
Parameter settings

Arrival probability	p=0.7
Number of service engine	N=10
NormalizedCPU speed of a baseline server	S=3.3
NormalizedCPU speed of a service engine	s=2.0:0.1:2.8
Parameters of energy model	K=1 α=3

C. Robustness of the REQUEST Algorithm

Although we estimate the time of transcoding jobs, this estimation may not be accurate, and the observation of the queue length is, also inaccurate. Therefore, it is necessary to study the robustness of the algorithm under the inaccurate queue length information. The robustness of the REQUEST algorithm, adding a random estimation error that is uniformly distributed, with the range of ±50% for the transcoding time. Consider as the relative error between the value of time average energy consumption (or time average queue length) using inaccurate queue length and the value using accurate queue length. The relative error of time average energy consumption (or time average queue length) is defined as the ratio between the difference due to the inaccuracy and the value using exact queue length. Plotting the errors of time average energy consumption and queue length in varying V. It shows that both errors are small. Hence, the REQUEST algorithm is robust to the transcoding time estimation.

VI. PERFORMANCE COMPARISON OF DISPATCHING ALGORITHMS

Here, we were comparing the performance of dispatching algorithms, including Round Robin, Random Rate, and REQUEST, under simulated traffic and real trace data.The Round Robin and Random Rate algorithms are illustrated as follows.

(1) *Round Robin:* Transcoding jobs are scheduled in a cyclical fashion among N service engines. It has time slices to assign each process in equal portions for circular order to handling all processes without any priority. It is simple and easy to implement. It can be also applied other scheduling process.

2) *Random Rate:* Transcoding jobs are dispatched to the ith service engine with the probability si/N i=1 si, which is proportional to the CPU speed of service engines. Round Robin and Random Rate algorithms are similar, in the sense that they attempt to make load balance among the service engines. These two algorithms are static and unaware of the arrivals, which limit their performance for achieving small energy consumption. The fundamental tradeoff between time average energy consumption and time average queue length for this REQUEST algorithm is more effective in minimizing the time.

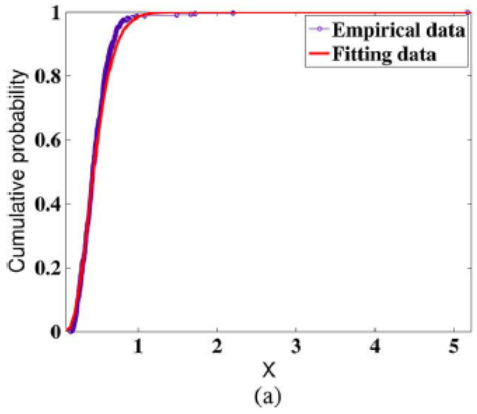
A. Simulated Traffic

The performance of the dispatching algorithm comes under simulated traffic. Connive fundamental tradeoff between the time average energy consumption and time length of the queue in the dispatching algorithm. The tradeoff of Random Rate and Round Robin algorithm is close to the request algorithm. The request algorithm is most effective in minimize the energy consumption and queue length. It choosing the different control variable V, the request algorithm is adaptive to stability between time average energy consumption and energy queue length. The energy efficiency process is to reduce amount of energy required for the services. The request algorithm has a larger time queue length than Round Robin and Random Rate, to maintain the queue length with in the margin. The Round Robin and Random Rate algorithms perform like to the proportional dispatching strategy.

B. Real Trace Data

At this time, we calculate and compare the performance of the dispatching algorithms by using real trace data. The trace data capture the video requests to a CDN node in China. We consider two periods of time, i.e., 6:00 PM– 11:00 AM and 11:00 AM–4:00 AM in a day. The data traffic of 11:00 AM–4:00 AM is lighter than that of 6:00 PM–11:00 AM. For every 30 min, we plot the time average energy consumption and time average queue length for these two periods in respectively. For the period of 6:00 PM–11:00 AM, since its traffic is heavier, it results in larger time average energy consumption and time average queue length for all dispatching algorithms. The request algorithm achieves short

time average queue length at high cost of time average energy consumption. The request algorithm is most energy efficient than the Round Robin and Random Rate algorithms. It may be maintains and manages the trade off between energy consumption and queue length of dispatching transcoding.



VII. CONCLUSION

To curtail the energy consumption by cloud service engines, and formulate the job-dispatching policy as an optimization problem under the framework of Lyapunov optimization. It characterized the energy–delay tradeoff and the robustness of the REQUEST algorithm. The simulation results shown in the REQUEST algorithm is more energy efficient than Round Robin and Random Rate algorithms. We find out the dispatching algorithms and it how to route transcoding jobs in the multimedia cloud. The insight of the cloud operator can dynamically tune the control variable of the REQUEST algorithm to reduce the energy consumption of maintaining the queue stability. We model the service engines as a set of parallel queues. Based on the *drift-plus-penalty* function, we propose an online algorithm that dispatches the transcoding jobs to the service engines to Reduce Energy consumption achieving the QUEue Stability (REQUEST).In the future, we make up a more general transcoding time model by considering the bit rate adaptation. In combine, we will take virtual machines into consideration for virtualized services for storing information temporary. Finally, we evaluate the process and performance of the proposed algorithm in the real multimedia platform.

VIII. REFERENCES

[1] Cisco Visual Networking Index: Forecast and Methodology, 2012–2017,2013.

[2] A. Vetro and C. W. Chen, "Rate-reduction transcoding design forwireless video streaming," in Proc. Int. Conf. Image Process., 2002, vol. 1,pp. I-29–I-32.

[3] Z. Li, Y. Huang, G. Liu, F. Wang, Z.-L. Zhang, and Y. Dai, "Cloud transcoder: Bridging the format and resolution gap between Internet videos and mobile devices," in Proc. 22nd Int. Worshop Netw. Oper. Syst.Support Digit. Audio Video, 2012, pp. 33–38.

[4] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," Synthesis Lectures Commun. Netw.,vol. 3, no. 1, pp. 1–211, 2010.

[5] A. Garcia and H. Kalva, "Cloud transcoding for mobile video contentdelivery," in Proc. IEEE ICCE, 2011, pp. 379–380.

[6] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[7] Y. Cui, X. Ma, J. Liu, and Y. Bao, "Energy-efficient on-demand streaming in mobile cellular networks," IEEE COMSOC MMTC E-LETTER, vol. 8, no. 5, pp. 1/49–49/49, Sep. 2013.