Scientific Journal Impact Factor (SJIF) : 3.518
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

# Toward Transcoding As A Service In A Multimedia Cloud Energy-Efficient Job Dispatching Algorithm

**Dr.S.Dhanalakshm[1]And R.Krishnaven[2]**

*[1]Head & professor Head of the Department And [2]Research Scholar Computer Science,*
*Vivekanandha College of Arts and Sciences for Women (Autonomous)*
*Elayampalayam, Tiruchengode india.*

**Abstract:** In this paper, we investigate the energy-efficient job dispatching algorithm for transcoding as a service (TaaS) in a multimedia cloud. We aim to minimize the energy consumption of service engines in the cloud while achieving low delay for TaaS. We formulate the job-dispatching problem as a constrained optimization problem under the framework of Lyapunov optimization. Using the *drift-plus-penalty* function, we propose an online algorithm that dispatches the transcoding jobs to service engines, with an objective to Reduce Energy consumption while achieving the QUEue STability (REQUEST).We first characterize the fundamental tradeoff between energy consumption and queue delay for the REQUEST algorithm numerically and obtain its performance bound theoretically. We study the robustness of the REQUEST algorithm, with numerical results indicating that the REQUEST algorithm is robust to the inaccuracy of estimating the transcoding time. We compare the performance of the REQUEST algorithm with the other two algorithms, i.e., the Round Robin and Random Rate algorithms. We show that by appropriately choosing the control variable, the REQUEST algorithm outperforms the Round Robin and Random Rate algorithms, with smaller time average energy consumption and time average queue length. The proposed REQUEST algorithm can be applied in cloud-assisted multimedia transcoding service.

**Keywords:** Energy efficiency, job dispatching, transcoding as a service.

## I. INTRODUCTION

The popularity of mobile devices, users have an increasing demand of online video consumption on devices. According to a Cisco VNI report [1], global Internet video traffic will contribute 69% of all Internet traffic in 2017: up from 57% in 2012. This trend of video consumption, however, may be hampered by the limited bandwidth and inherent nature of stochastic wireless channels (e.g., multipath fading and shadowing effects), which can degrade the user's experience while watching videos. Transcoding technology is introduced to adapt the videos according to the available bandwidth or different users' requirements.

Basically, a content provider can transcode the same video into multiple rates or multiple formats for users' need. In addition, the resolution size of a video can be reduced such that users can view the video smoothly over the network.

However, such a transcoding process is computation intensive for the content provider. It is a challenge for the content provider to maintain the low delay for transcoding when many requests arrive. Therefore, a large-scale platform should be designed to support the transcoding process

Cloud computing, due to its elasticity of resource allocation, offers a natural way to process a very large number of transcoding jobs. A large number of servers in the cloud can perform transcoding jobs on behalf of the content provider. In this case, the content provider can benefit from the cloud for video consumption from users. This has become an opportunity to deliver transcoding as a service

 

(TaaS). A generic cloud-assisted transcoding system is shown in Fig. 1(a). Particularly, users request a content with specific requirement (e.g., bit rate and resolution size), which is determined by the physical capability of the devices and the available bandwidth. If a particular content is available at the content provider, the content can be rendered immediately. Otherwise, the content provider will send a transcoding job to the cloud to cater for the requirement of users. In the cloud, there is adispatcher at the front end and a large number of service engines at the back end. The arriving transcoding job is routed by the dispatcher and completed by one service engine in the cloud.

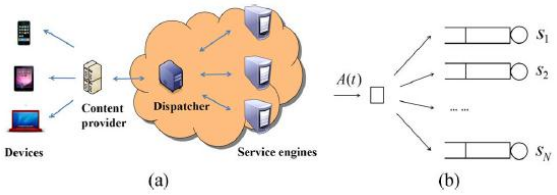


*Fig.1: Overview of a cloud-assisted multimedia transcoding platform.*

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.

## II. RELATED WORK

Prior works have investigated transcoding in distribute systems. In, tasks are scheduled for a cluster-based web server to process 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. Those two works did not investigate the robustness of the scheduling algorithms for the case that the estimation model is not accurate. In this paper, our proposed algorithm is robust to the inaccuracy of the estimated time. Another line of research leverages cloud computing to enhance the performance of transcoding. In and, a Hadoop-based cloud fortranscoding media content is utilized, which can greatly improve encoding times. In, a cloud transcoder to bridge the gap between videos and mobile devices is proposed, reducing the transcoding burden on mobile devices. In, a 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, a load-sharing algorithm in a transcoding cluster is provided, and in, 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, the cost-efficient virtual machine provision for video transcoding, is considered. In, mechanisms for allocation and deallocation of virtual machines to video transcoding servers is provided.In addition, and attempted to minimize the transmission energy on the mobile devices. However, neither of them considered the energy consumption of servers for transcoding. This paper aims to minimize the energy consumption in the cloud for TaaS while maintaining the queue stability. The tradeoff between energy consumption and queue stability is characterized under the framework of Lyapunov optimization.

## III. PROBLEM STATEMENT

### A. Exiting Model

As mentioned in the previous sections Transcoding is a computationally heavy task. Especially, small devices such as routers and mobile phones are not capable of handling it alone for on demand streaming requirements with acceptable user experience. The main objective of this thesis is to investigate existing distributed computing infrastructures and how they are being utilized from the point of view of the video stream processing and propose a way or architecture on how to utilize such

infrastructures in an efficient manner. The investigation will begin with studying and finding the right system components that are required. In addition to this a minimal version of such system needs to be implemented on a selected platform for the purpose of further experimentation and performance analysis of the components.

Video content providers can transcode the same video into multiple rates or multiple formats for users' need. In addition, the resolution size of a video can be reduced such that users can view the video smoothly over the network.

However, such a transcoding process is computation intensive for the content provider. It is a challenge for the content provider to maintain the low delay for transcoding when many requests arrive. Therefore, a large-scale platform should be designed to support the transcoding process.Load balancing.

In any distributed system load distribution is a challenging task. Without proper load balancing a distributed implementation of a transcoder might perform badly or even worse than its strictly sequential counterpart. The main challenge of load distribution from the perspective of transcoding comes directly from the video stream structure which contains variable data in terms of processing time dependent only on the type of data contained.

### B. Proposed Model:

We propose an online algorithm that dispatches the transcoding jobs to the service engines to Reduce Energy consumption while achieving the QUEue STability (REQUEST).

- 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.
- We study the robustness of the REQUEST algorithm. Numerical results show that, given the inaccuracy of estimating the transcoding time, the error of the time average energy consumption and queue backlog is small. Therefore, 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 results show that by appropriately choosing the control variable, the REQUEST algorithm outperforms the other two algorithms, with smaller time average energy consumption while achieving queue stability.

## IV. PROBLEM STATEMENT

### Arrival model

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

### Queueing Model

We model the service engines as a set of queues, as shown in Fig. 1(b). To characterize the dynamics of these queues, we define queue length $\mathbf{Q}(t)$ as the unfinished transcoding time of jobs in each service engine at time slot $t$, i.e., $\mathbf{Q}(t) = \{Q1(t), Q2(t), \ldots, QN(t)\}$. The queue of the $i$th service engine evolves according to $Qi(t + 1) = \max[Qi(t) - \tau, 0] + Ai(t)\mathbf{1}\{u(t)=i\}$

### Energy Consumption Model:

We consider each service engine as a physical machine. Particularly, we only consider the computation energy consumption in the service engine, which is a dominant term for the energy consumption in the distributed servers. As such, we ignore other sources of energy consumption in the service engine, e.g., memory and network. We assume that each service engine operates in a constant CPU speed when processing transcoding jobs.

 

Its resulted energy consumption is assumed to be a function of CPU speed

### A. Problem Formulation

Intuitively, if the dispatcher routes the transcoding job to the service engine with the least queue backlog, it can reduce the delay for the transcoding job; however, it would incur large energy consumption if many transcoding jobs are dispatched

to service engines with fast CPU speed. If the dispatcher routes many transcoding jobs to the service engine with the slow CPU speed, it can reduce the energy consumption; however, it would make the queue arbitrarily long and incur long delay. Therefore, we consider the tradeoff between energy consumption and time delay.In this paper, we aim to minimize the long-term time average energy consumption subject to the constraint that time average queue length should not go to infinity.

## V. Performance Comparison of Dispatching Algorithms

Here, we compare 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.

2) *Random Rate:* Transcoding jobs are dispatched to the *i*th service engine with the probability *si/ Ni*=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. However, these two algorithms are static and unaware of the arrivals, which limits their performance for achieving small energy consumption.

## VI. Conclusion

We investigated dispatching algorithms on how to route transcoding jobs in the multimedia cloud. To minimize the energy consumption by cloud service engines, we formulated the job-dispatching policy as an optimization problem under the framework of Lyapunov optimization. We characterized the energy–delay tradeoff and the robustness of the REQUEST algorithm. The simulation results showed that the REQUEST algorithm is more energy efficient than Round Robin and Random Rate algorithms. The insight is that the cloud operator can dynamically tune the control variable of the REQUEST algorithm to reduce the energy consumption while maintaining the queue stability.In the future, we will build up a more general transcoding time model by considering the bit rate adaptation. In addition, we will take virtual machines into consideration for virtualized services. Finally, we will evaluate the performance of the proposed algorithm in the real multimedia platform.

## REFERENCE

[1] F. Jokhio, A. Ashraf, S. Lafond, I. Porres, and J. Lilius, “Prediction-based dynamic resource allocation for video transcoding in cloud computing,” in Proc. 21st Euromicro Int. Conf. PDP, 2013, pp. 254–261.

[2] 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.

[3] S. Ko, S. Park, and H. Han, “Design analysis for real-time video transcoding on cloud systems,” in Proc. 28th Annu. ACM Symp. Appl. Comput., 2013, pp. 1610–1615.

[4] J. Guo and L. Bhuyan, “Load sharing in a transcoding cluster,” in Proc.Distrib. Comput.-IWDC, 2003, pp. 330–339.

[5] H. Sanson, L. Loyola, and D. Pereira, “Scalable distributed architecture for media transcoding,” in Proc. Algorithms Archit. Parallel Process., 2012, pp. 288–302.

[6] A. Ashraf, F. Jokhio, T. Deneke, S. Lafond, I. Porres, and J. Lilius, “Stream-based admission control and scheduling for video transcoding in cloud computing,” in Proc. 13th IEEE/ACM Int. Symp. CCGrid, 2013, pp. 482–489.

[7] A. Ashraf, “Cost-efficient virtual machine provisioning for multi-tier web applications and video transcoding,” in Proc. 13th IEEE/ACM Int. Symp. CCGrid, 2013, pp. 66–69.

 