ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

VIDEO TRANSCODE SERVICE PLATFORM WITH WORKLOAD PREDICTION

[1]GAODE QIN, [2]GAOJIN WEN
[1]Shenzhen Polytechnic, Shenzhen 518055, Guangdong, China
[2]National Laboratory of Pattern Recognition, Institute of Automation of the Chinese Academy of Sciences,
95 Zhongguancun East Road, 100190, Beijing, China

ABSTRACT

With the rapid expanding of the integration of telecommunications networks, cable TV networks and the internet, the efficient and energy-saving video transcode service platform is in great demand. Such a platform together with workload prediction is proposed in this paper. A new accurate multi-scale prediction method is used to predict the execution time of transcode tasks in this platform. Special distribution and management scheme is applied to serve the purpose of energy optimization. With several experiments, it is proved that the proposed platform can be used to predict the workload of the transcode tasks accurately and save the energy efficiently by task scheduling optimization.

Keywords: *Multi-scale Prediction, Video Transcode, Task Scheduling*

1. INTRODUCTION

Transcode high-bit-rate video stream into low bit rate video stream, by the demand of equipment and software in different network bandwidth and play terminal is one of the most important applications of video transcoding [1,2,3,4,5]. Therefore, the Harley in 2009 introduced workflow-based distributed transcoding system Carbon Server, the system uses a workflow and high-speed video transcoding services. In 2010, Microsoft released a video transcoding engine (Transform Manager Transcode Engine) based on the cluster system, the system software has good scalability, support priority task scheduling and fault tolerance mechanism.

By far, video transcoding systems do not predict the task workload of video transcoding in order to improve transcoding performance. Thus, this paper proposes a video transcoding service platform system based on workload prediction, which uses a new type of multi-scale prediction method to estimate the execution time of transcoding tasks, and also perform optimization of transcoding task scheduling optimization based on the predicted time. Under the premise of the normal operation, the system aims at optimize the energy consumption, allocates and manages data center nodes to maximize the efficiency of the nodes.

2. ARCHITECTURE OF VIDEO TRANSCODING SERVICE PLATFORM SYSTEM

Video transcoding service platform system consists of one management node and many compute nodes sharing storage. System software running on the management node is responsible for the entire platform system management and scheduling, while video transcoding program on compute nodes is responsible for the execution of the transcoding task.

Video transcoding service platform system software consists of the following seven modules illustrated in Figure 1:

1) video data management module, is mainly responsible for the video data import, export, backup and delete operations;

2) task planning modules generate video transcoding batch task primarily according to the user's requirements (video data, processing time, format transcoding parameters);

3) log module records transcoding task execution time and node status information;

4) workload prediction module, is mainly responsible for generating task workload prediction model. In the next section, we will discuss the multi-scale workload prediction algorithm in details;

Journal of Theoretical and Applied Information Technology
15th December 2012. Vol. 46 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5) task scheduling optimization is mainly responsible for making task scheduling program based on the task workload of prediction model and the status of nodes;

6) node monitoring module monitors the real-time performance of nodes;

7) task scheduling execution module is mainly responsible for scheduling tasks according to the real-time status of the nodes which includes node wake-up and sleep operation and the task migration of fault node.

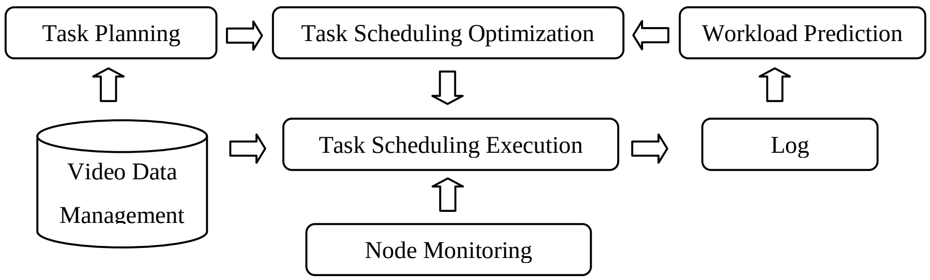

Figure 1: Architecture of video transcoding service platform system

3. MULTI-SCALE WORKLOAD PREDICTION ALGORITHM

After performing one video transcoding task, you can get a historical log data vector x of which the component are main parameters which affect the video transcoding time: input video encoding format, output video coding format, input video resolution, output video resolution, video length, audio format, frame rate, the execution time, etc. According to the different input and output video formats, we can classify the historical log data. Therefore, we can predict the task execution time of each class of tasks.

Algorithm 1: multi-scale workload prediction algorithm

Input: historical log data of task $\{x_i\}_{i=1}^n$, the maximum matching scale layer MaxLevel = 3 or 4, the prediction error accuracy $\varepsilon = 10e^{-6}$

Output: sequence Ω of multi-scale prediction model

Step 1: Set current matching scale CurLevel=1, $\Omega = \theta$

Step 2: Analyze using a variety of time-series regression analysis method of which the procedure is as follows:

(1) Use four models for workload prediction of $\{x_i\}_{i=1}^n$, general autoregressive model $AR(p)$, the moving average model $MA(q)$, autoregressive moving average model $ARMA(p,q)$ and autoregressive moving average model $ARIMA(p,q)$. Let $\{y_i^j\}_{i-1}^n$ be the predictive value of each model where $j = 1,2,3,4$ represents the model;

(2) Calculate the absolute error of predicted results from the models in (1). The formula for the j-th model is $E_j = \sum_{i=1}^{n} \left| y_i^j - x_i \right|^2$;

(3) Select the model of which relative error is minimum to be the final predictive model in the current scale. Suppose the k-th model bears minimum relative error. Add it to the sequence Ω (Set $\Omega = \Omega \cup \{k\}$).

Step 3: Update the input data of current time sequence, $x_i = x_i - y_i^k$, where $i = 1,...,n$. Compute the error of current matching $\omega = \sum_{i=1}^{n} \|x_i\|^2$

Step 4: IF ($\omega < \varepsilon$) THEN goto Step 6; ELSE goto Step 5

Step 5: IF (CurLevel < MaxLevel) THEN goto Step 2; ELSE CurLevel = CurLevel + 1 and goto Step 6

Step 6: Output Ω

4. TASK SCHEDULING OPTIMIZATION

If we can get relative accurate predicition of the task workload, then under the premise of the video transcoding task execution time, we can achieve an

Journal of Theoretical and Applied Information Technology
15th December 2012. Vol. 46 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

energy saving target using the minimum node to complete the task of transcoding.

Assumes that a set of transcoding tasks need to process n batches video data, a high-definition video input data needs to be converted into k formats. This group transcoding tasks workload is $\sum_{i=1}^{n}\sum_{j=1}^{k} t_i^j$.

If the user requires the video transcoding task execution time is T, it needs at least $N = \left\lceil \sum_{i=1}^{n}\sum_{j=1}^{k} t_i^j \Big/ T \right\rceil$ nodes. In addition, transcoding tasks scheduling constrains that the video files processed by N nodes are not the same anytime. We can amend the Best Fit Decreasing method in [6] to solve the video transcoding task scheduling problems. The procedure is as follows:

Algorithm 2: constraint best fit decreasing algorithm

Input: a sery of video transcoding tasks $t_1^1,\ t_1^2,\ \ldots, t_1^k,\ t_2^1, t_2^2, \ldots, t_{n-1}^k, t_n^2, \ldots, t_n^k$

Output: the number of nodes Num , task assignment Nds ;

Step 1: sort the transcoding tasks decreasingly $t_1 \geq t_2 \geq t_3 \geq \ldots \geq t_n$ and set $Nds = \theta$;

Step 2: find the value of Num and the task list on each node $\{Task_j\}_{j=1}^{Num}$ by best fit method;

Step 3: Insert the tasks of the first node into Nds ;

Step 4: For j=2 to Num

{ record the task in $Task_j$ by t_i^{sub} , $1 \leq i \leq s_j$, Pos=1; s=0;

While ($s != s_j$)

{ For i=1 to s_j

{ If (t_i^{sub} does not conflict with Nds)

{ $Nds = Nds \cup \{(t_i^{sub}, j, pos)\}$;

pos=pos + t_i^{sub} ; s++; }

else i = s_j ;

} } }

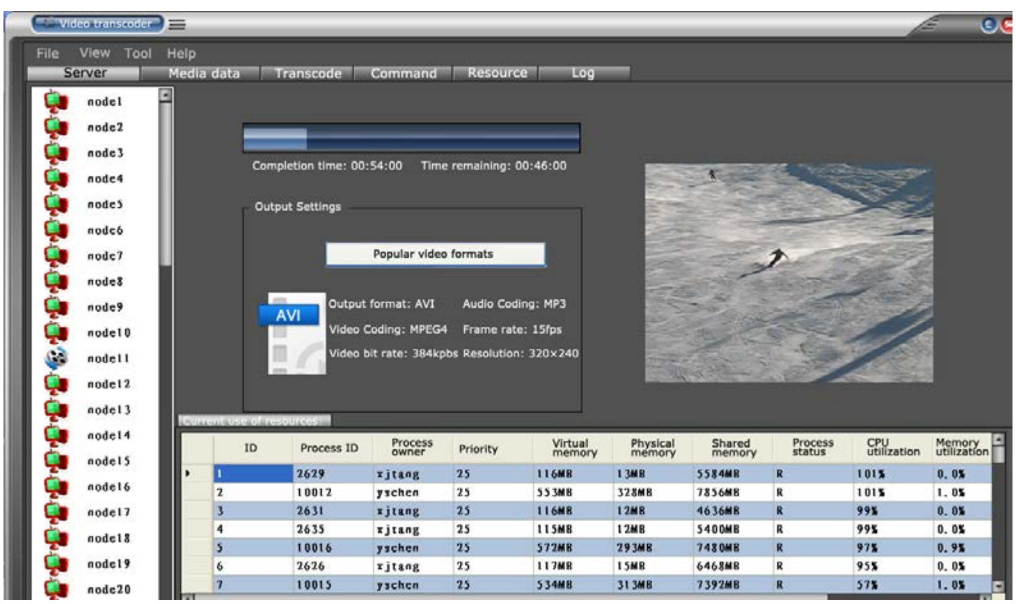

Figure 2 The Video Transcoding Service Platform Monitoring Interface

Journal of Theoretical and Applied Information Technology
15th December 2012. Vol. 46 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

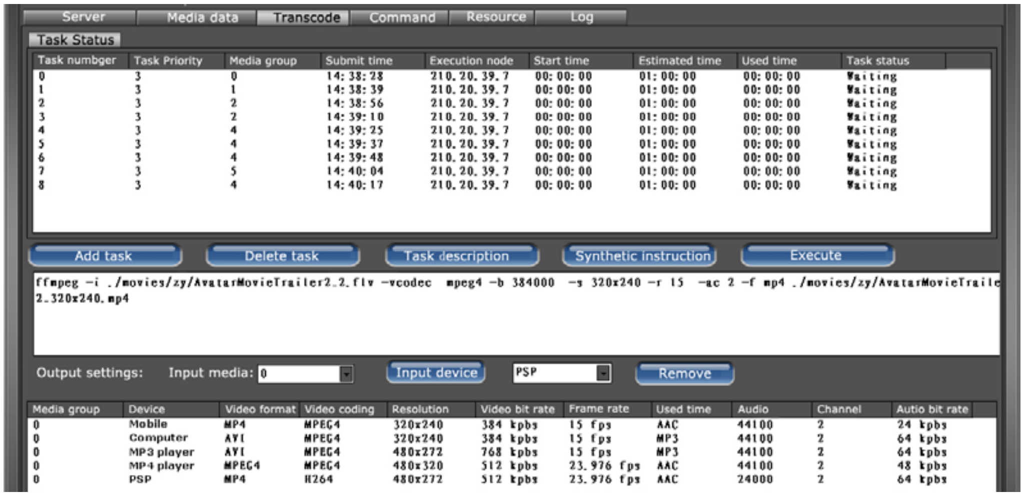

Figure 3: The Video Transcoding Task Developing Interface

5. SYSTEM IMPLEMENT AND RESULTS ANALYSIS

5.1 Video Transcoding Service Platform

We give one C++ implementation of our video transcoding service platform system. The hardware system comprises one management node and four compute nodes. Open source video transcoding program FFMPEG is deployed on the compute nodes. Figure 2 and Figure 3 show the interface of our system.

5.2 Task Workload Prediction

We implement the multi-scale task workload prediction module algorithm. We use 80 groups of high-definition video data transcoding job as a testing task. We extract the input / output video formats, stream frames, frame size, CPU occupancy rate, as the analysis of elements of the execution time.

The experiment employs 70 batches of HD video data for modeling and 10 batches of data for model validation. Compared with Weka [6] in four different algorithm (decision tree, M5P, Gaussian process, linear regression), our method can achieve better prediction shown in Table 1.

Table 1 : Comparison Of Video Transcoding Task Workload Prediction Results

Item	Multi-scale prediction	Decision tree	M5P	Gaussian process	Linear regression
Relation coefficient	0.99	0.97	0.96	0.95	0.87
Average absolute error	5.85	12.68	16.06	33.78	27.10
RMS error	10.71	18.62	21.53	42.18	38.51
Relative error	9.05%	19.63%	24.87%	52.28%	41.95%
RMS relative error	13.9%	24.16%	27.94%	54.73%	49.97%

5.3 Task Scheduling Optimization

We also implement and test constraint best fit decreasing algorithm (Algorithm 2). Figure 4 is the task workload prediction map for the task of converting 80 groups of high-definition video data into five different media formats. The total workload is 16051 minutes. Assumes that the user requires the video transcoding task execution time is 1500 minutes. Then, it needs $\lceil 16051/1500 \rceil = 11$ nodes. The task scheduling table obtained by Algorithm 2 is shown in Figure 5. After performing many tests, it shows that Algorithm 2 is able to deal effectively with the task scheduling problem of video transcoding service platform.

Task scheduling plan is given in Figure 5. The longitudinal length of the lattice in the figure stands for the task execution time and the number in lattice is the media data processing label. By Figure 5, it appears that there are no nodes processing the media data with same label, simultaneously.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

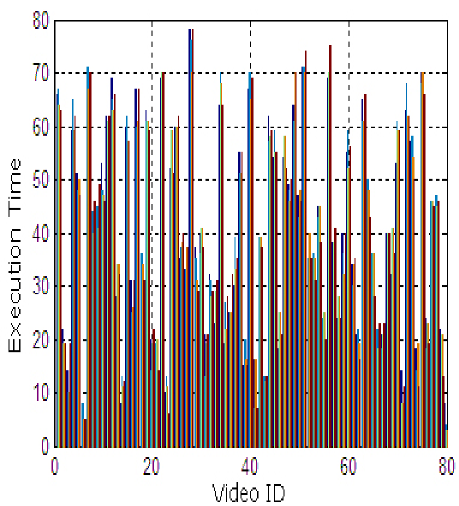

Figure 4: The Task Workload Prediction

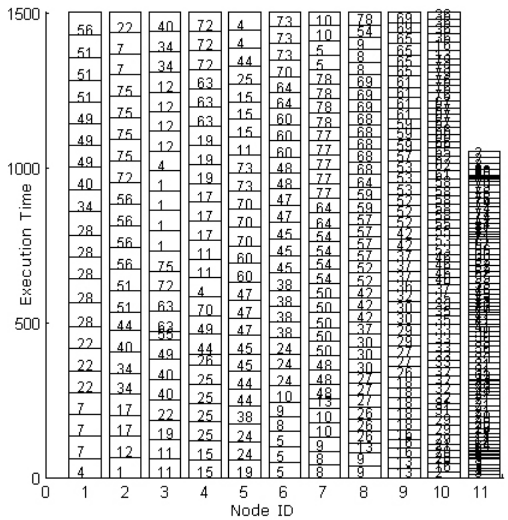

Figure 5: The Task Scheduling Plan

6. CONCLUSION

This paper presents a video transcoding service platform system architecture. Its virtue is to adopt one multi-scale workload prediction algorithm which can accurately predict video transcoding task workload. It helps to improve the video transcoding service platform in efficiency of resource usage and energy consumption.

REFERENCES:

[1] Xin Jin, "Hilbert Transform-Based Workload Prediction and Dynamic Frequency Scaling for Power-Efficient Video Encoding", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions*, Vol.31, No. 5, 2012, pp. 649-661

[2] Bao Sheng, Duan Baotong, Shao Fengjun, "Research and Application of Real-Time Transcoding Based on Cloud-Computing Technology in Triple-Play", *Telecommunications Science*, Vol. 27, 2011, pp.12-16.

[3] Wang Bo, Liu Dichen, Li Min, "A Mixture Video Transcoding Method for H. 264", *Journal of Image and Graphics*, Vol. 26, 2009, pp.1961-1964.

[4] Han Zhen, Hu Ruimin, Chen Hao, Li Ming, "Error Resilient Video Transcoding Based on Layered Rate Distortion Optimization",*Chinese Journal of Computers*, Vol. 32, 2009, pp. 1152-1161.

[5] Zhou Yiran Chen Yaowu, "Error-resilient video transcoding based on adaptive redundant slice insertion", *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, Vol. 39, 2011, pp. 6-10.

[6] Yue, Minyi, "A simple proof of the inequality FFD(L) ⩽ (11/9)OPT(L) + 1, for all L, for the FFD bin-packing algorithm", *Acta Mathematicae Applicatae Sinica*, Vol. 7, 1991, pp. 321-331.