PARALLELIZING VIDEO TRANSCODING USING MAP-REDUCE-BASED CLOUD COMPUTING

Feng Lao, Xinggong Zhang and Zongming Guo
Institute of Computer Science & Technology
Peking University, Beijing 100871, P.R. China
Email: {laofeng, zhangxg, guozongming}@pku.edu.cn
Outline

1. INTRODUCTION
2. SYSTEM ARCHITECTURE
3. PROBLEM FORMULATION
4. MAX-MCT ALGORITHM
5. EXPERIMENT
6. CONCLUSION
Recent years, there has been a growing demand for high quality video, which leads to advances of coding technology, such as H.264, MPEG-4 and MPEG-2 and so on.

And various environments usually require different coding formats.
INTRODUCTION

- This results in the demand of fast transcoding.

- However, due to the complexity of video coding, fast transcoding remains a problem to be explored.
INTRODUCTION

- There have been many efforts devoted to parallel transcoding over multi-core processor, such as [1] [2] [3].

- But due to specified hardware, the parallel transcoding over multi-core processor is hard to extend.
Cloud computing, as an emerging technology, can utilize computing power of thousands of computers.

Cloud computing consists of a cluster of distributed computers.
Since the computers can be heterogeneous, cloud computing is extendable and relatively inexpensive.

Map/Reduce is a distributing cloud computing model.
INTRODUCTION

Fig. 1. Cloud Computing Architecture
Moreover, when the transcoding time is in proportion to segment complexity, **Min-Min algorithm** is equal to **minimal complete time (MCT) algorithm**.

The MCT algorithm assigns segments according to descending complexity order.
We formulate the scheduling as an NP-hard problem.

Considering overhead to launch sub-tasks, we propose a heuristic task scheduling algorithm, named Maximizing Minimal Complete Time (Max-MCT), which includes two procedures: virtual knapsack and MCT procedures.
Fig. 2. System Architecture
To insure the independency of the segments, video sequence should be divided in between GOPs.

Moreover, the content of each segment is also various.

Therefore, the complexity of segments is heterogeneous.
As present above, we are given $n$ different segments,

$$J = (1,2,3\ldots,n)$$

- with different complexity, $C = (C_1,C_2,C_3\ldots,C_n)$
- Each segment must be processed without preemption until its completion.
- We also have $m$ computers with different capacity,

$$P = (P_1,P_2,P_3\ldots,P_N)$$

- And the task-launching overhead is $T_{\text{overhead}}$. 

PROBLEM FORMULATION

- The transcoding time is proportional to segment complexity and inversely proportional to computer capacity.

- So that the transcoding time spent for segment $i$ on computer $j$ can be computed as $t_{ij} = c_i/p_j + T_{overhead}$.

$$f_{S_j} = \sum_{i \in S_j} t_{ij} = \sum_{i \in S_j} c_i/p_j + |S_j| \times t_{overhead}$$
MAX-MCT ALGORITHM

- A. Virtual knapsack procedure

- At first, we are given \( m \) computers and \( n \) segments. We can estimate the low-bound as the average finish time:

\[
f^* = \frac{\sum c_i}{\sum p_j + t_{overhead} \times n/m}
\]

- we can treat the computers as virtual knapsacks with a volume \( V_i = p_i \times f^* \)
- And there are \( n \) items weighting \( c_i + \text{overhead} \).
MAX-MCT ALGORITHM

Fig. 3. Virtual Knapsack
MAX-MCT ALGORITHM

B. MCT procedure

Here, we employ MCT algorithm to handle the residual pieces.

Then the computer with the minimal complete time is chosen.

It continues until all the residual segments have been assigned.
C. Algorithm Analysis

Now, we analyze the complexity of our Max-MCT algorithm.

- Sorting the segments according has complexity $O(n \log n)$.
- The virtual knapsack procedure has complexity $O(n)$ and the complexity of MCT algorithm is $O(nm)$.
- So our algorithm has a low complexity $O(n \log n)$. 
EXPERIMENT

- Thus we employ Matlab to conduct simulation experiments to evaluate different scheduling strategies.

- Generally, we create 8 computers, with capacity ranging from 5 to 15, and 300 video segments whose complexity ranges from 300 to 900.

- And the task-launching overhead is 10.
For each situation, we conduct 1000 experiments and pick the average as an output.

Here we mainly evaluate the Max- MCT algorithm against MCT algorithm.
Exceeding time (s)

Number of video segments

(a) different numbers of segments
(b) different numbers of computers
(c) different range of complexity
Fig. 5. The number of segments in different computers
CONCLUSION

- In this paper, we investigate the fast transcoding problem and present a Map-Reduce-based cloud transcoding system.

- To reduce complexity, we propose a heuristic algorithm named Max-MCT with two procedures.

- We also conduct various simulation experiments to verify that our algorithm outperforms the exiting algorithms.