Acceleration of Video Conversion on the GPU based Cloud

Prof. Sandip M. Walunj[1], Akash Talole[2], Gaurav Taori[2], Sachin Kothawade[2]

[1] *Professor, Department of Computer Engineering, Sandip Institute of Technology and Research Centre, Maharashtra, India*

[2] *Students, Department of Computer Engineering, Sandip Institute of Technology and Research Centre, Maharashtra, India*

Abstract - *Since past few years it is being seen that the conversion of video file is done in order to make it compatible with different devices such as personal computers, mobile phones, tablets and Smart TV. The traditional video converter uses CPU to convert the video file from AVI to MPEG and may provide limited user access at a time and also takes time to convert videos. Hence we have developed Graphics Processing Unit based video converter in the cloud.*

During initial testing phase for a 2mb avi file the conversion rate is shows an efficiency of 0.0012%. Though it is less in percentage but with more optimized code it is expected that it can achieve higher efficiency rate.

Key Words: *AVI files, Cloud, Parallel Processing, Compute Unified Device Architecture (CUDA), Graphics Processing Unit (GPU), MPEG files , Video Conversion*

1. INTRODUCTION

The front end of the proposed system has a web user interface from where the user can upload and download the video file and at the backend combination of GPU-Cloud structure of the video converter is present.

The user will upload the AVI video file to the cloud server using the web interface as shown in the figure 1 and then further process of conversion is carried out by GPU using its parallel threads and after the successful conversion to MPEG file, the file is available for download.

Cloud Computing: The architecture that refers to the group of geographically located machines, configured as data storage & sharing network[1][2].

There are companies that have developed their own cloud services[1][3] for web search, emails, social networking, e-commerce, and more as it has various characteristics such as easily maintainable, performance and productivity, cost efficient, scalable, device independent etc. Virtualization with GPU and cloud is an important and useful aspect of the modern world as it allows multiple user and applications to use the resources and services of the cloud with much less computational time.

Graphics Processing Unit (GPU): GPU is an electronic circuit used to manipulate and accelerate the processing of the data, based on given input. It uses 'n' core parallel structure compared with CPU serial structure. It can also be used for general purpose computing such as molecular dynamics simulation with replica exchange method[1][3].

During the early years, GPU were not stable in the cloud and its performance was poor but with the recent technology and study, it is possible[1][3][5][6] to use GPU in cloud.

Compute Unified Device Architecture (CUDA): It is a parallel executing programming structure, which is founded by NVIDIA for utilizing the potential of GPU[7] in general purpose computing. The use of CUDA in GPU allows developer to perform parallel algorithm computation with much ease.

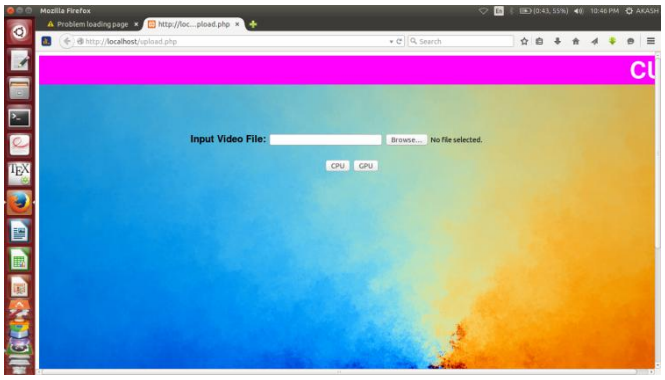


Fig -1: Web Interface

2. RELATED WORK

The earlier methods/systems/application/algorithms based on the GPU based Cloud is discussed below.

2.1 Graph Processing

All the data over the cloud is kept using the graph data structure and sometimes the graph becomes so large that its processing speed is not as expected and execution

becomes difficult. To increase the processing speed and simplify the execution, GPU-Cloud is used. The system, uses the cloud services provided by Amazon EC2 with 8 nodes and implements the following[4]:

1. Fine grain API: To use parallel threads of GPU.
2. Graphs partition for load balancing on GPU-CPU architecture.
3. Another runtime system for transparent memory management on GPU and scheduling algorithm to improve the throughput of graph tasks.

After complete execution, the system shows a 50% improved performance.

2.2 Ray-Tracing application versus Gaming Applications

In the field of gaming, ray-tracing applications are used to generate images by tracing the path of the light through pixels. Gaming applications are set of graphical running images which are controlled by a certain set of mouse or keyboard keys. The system[5] developed to compare these two applications had found that, the ray tracing application has performed better when used in virtualized GPU in cloud whereas gaming application performs poorly.

2.3 Classical Image Algorithms using CUDA

Processing of classical image algorithm such as equalizing the histogram, removal of cloud, DCT encode and decode, edge detection and many more algorithms can be done using GPU. So the system[8] developed to perform this operation shows the following:

1. 40x more speed in execution of histogram,
2. 79x more speed for removing clouds
3. 8x more speed for DCT and
4. 200x more speed for edge detection

2.4 Structural Comparison of Protein Binding Sites

The system[9] was originally written in Java, which can only run on CPU. Now it is revised under OpenCL language, so as to make it runnable using parallel cores of GPU set of instances of cloud services of Amazon EC2. This is the new set of implementation of SEGA and shows an notable computation time of three weeks.

2.5 Reducing Database Search Workload

Earlier, Smith-Waterman algorithm was used for database searching but it takes much more time, so a new system[10] is developed which uses CUDA programming to improve the computations of searching. A frequency based filter method is used to deal with all the unnecessary searches and comparisons. The system shows 41% improved performance by reducing the unnecessary sequence alignment and achieved a 76 times more speed ratio than single GPU for overall search time.

2.6 Dynamic Sharing

The authors[11] have proposed a framework known as gCloud, which provide GPU-cloud services, to the user as per their demands. The system shares GPU resources with ease among different applications running in parallel from different cloud users and it shows an improved use of GPU i.e. 56.3% and reduce the overall application time i.e. by 10.3%[10], compared with round robin based consolidation policy.

3. METHODOLOGY

3.1 Mathematical Model

We can represent the system S as a set of 5 tuples, as shown below[1]:

$$S = \{I, O, G, Q, T\} \quad (1)$$

Where,

I = Input (avi file)

O = Output (mpeg file)

G = GPU Cores

Q = Queue

T= Time of conversion

The proposed system can be used by 'n' users, hence there may be 'n' inputs for them which can be represented as shown in equation (2)

$$I = \{ I_i \mid \text{where } 0 < i < n\} \quad (2)$$

As the system as 'n' inputs, it will have 'n' outputs too as shown below in equation (3)

$$O = \{ O_i \mid \text{where } 0 < i < n\} \quad (3)$$

The relationship between Input and Output can be shown as in figure 1. It has a one to one relationship and hence we can say,

$$I \propto O \quad (4)$$

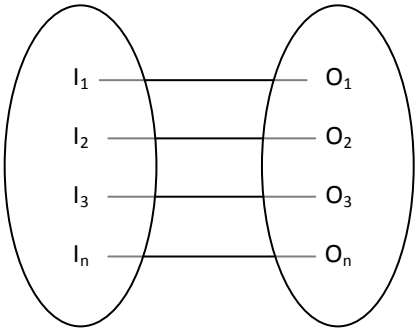


Fig.- 2: One-to-one relationship between Input set and Output set[1]

Now Q is the data structure Queue, which is used to manage the priority of video files being uploaded in the cloud.

Hence figure 2 can be expanded as shown in figure 3.

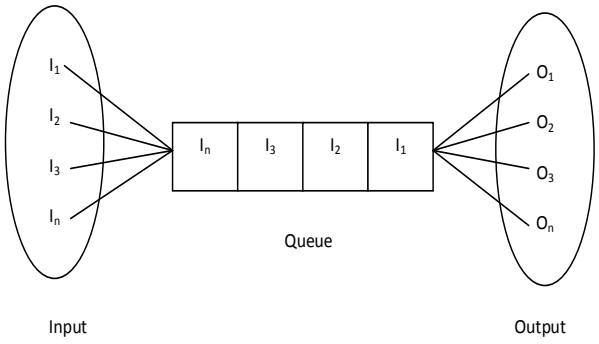


Fig.- 3: Input/output with Queue[1]

'G' can be represented as set of GPU cores required to convert a particular video file from the input set. Hence figure 3 can be expanded as shown in figure 4.

T' is used to represent as set of time required to convert the video file. It can be represented as shown in equation (5)

$$T = \{ T_i \mid \text{where } 0 < i < n\} \quad (5)$$

Let us consider 't_0' is the time taken by the video file(avi file) to reach the GPU present in the cloud and 't_1' is the time taken by the video file(mpeg file) to reach the terminal back. Therefore time to reach the cloud and to come back can be represented as:

$$t_0 + t_1 = T_c \quad (6)$$

Where T_c is the total time taken to reach the cloud and to come back to the terminal.

We will consider T_c equal to 1 as it depends on the internet speed which is different for different networks.

$$T_c = 1 \quad (7)$$

When we convert a number of video files using the CPU, it converts it serially, whereas GPU converts it parallel. Mathematically, if T_{CPU} is the complete time taken by CPU to convert 'n' number of video files and T_{GPU} is the complete time taken by GPU to convert the same number of video files, then we can say from equation (5):

$$T_{CPU} = T_1 + T_2 + T_3 + \ldots + T_n \quad (8)$$

$$T_{GPU} = (1/T_1) + (1/T_2) + (1/T_3) + \ldots + (1/T_n) \quad (9)$$

If we consider only two input files then, equation (8) can be represented as

$$T_{CPU} = T_1 + T_2 \quad (10)$$

and equation (9) can be represented as

$$T_{GPU} = (T_2 + T_1)/(T_1 * T_2) \quad (11)$$

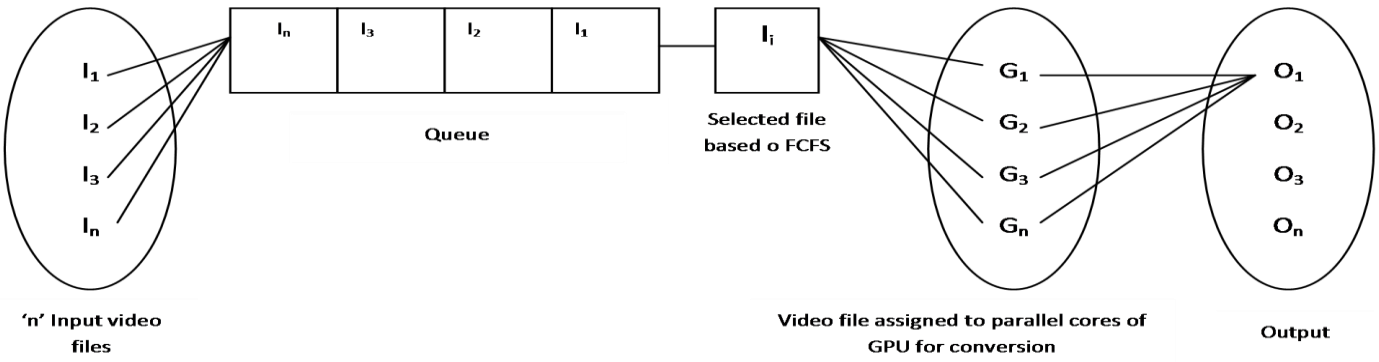


Fig-4: Expanded Diagram with GPU Cores[1]

3.2 Activity Diagram

1. As shown in figure 5, we can describe the flow of the video file in the proposed system[1].
2. The user or actor initiates the process by uploading a video file (avi file) as input from the UI provided for the proposed system.
3. It goes to the local cloud storage.
4. Then through the network nodes it goes to local cloud storage.
5. From local cloud storage it goes to the server storage where the GPU is present.
6. Using that GPU the file is converted by parallel threads and then the converted file is given back to the user.
7. From the UI, the user will download the converted video file (mpeg file).

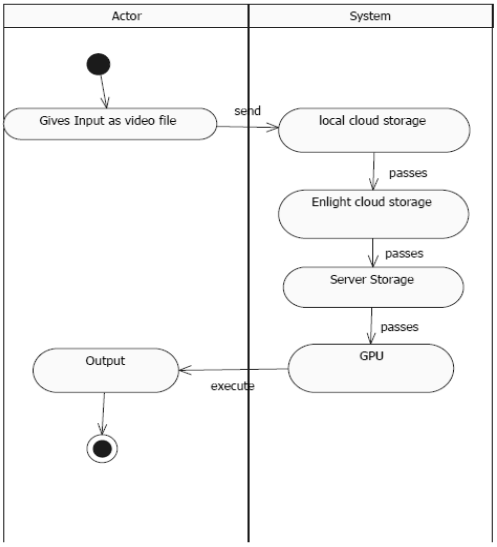


Fig-5: Activity Diagram of the Proposed System[1]

3.3 System Architecture

The system architecture contains 'n' number of devices which may or may not have graphics card installed in it. An interface between the cloud and the terminal is established in order to communicate and upload video files for conversion via the GPU present in cloud[1].

From Figure 6, we can explain that first of all the users will upload the video file (avi file) from the provided web interface. The file goes to the internet cloud and from there it goes to the local cloud where GPU is present.

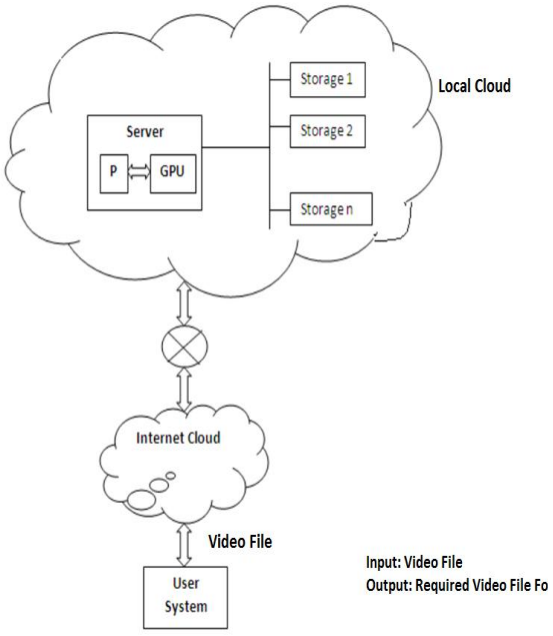


Fig-6: System Architecture[1]

The file get stored in the server storage space and from there, using queue (on the basis of FCFS), conversion process start for each video file. After each successful conversion, the converted video file (mpeg file) is send to the user interface with a download link to download it.

4. SOFTWARE AND HARDWARE REQUIREMENT

The following numbers of software and hardware is required for the proposed system.

4.1 Software Requirements

1. eNlight Cloud(Local Cloud)
2. CUDA 6 Toolkit

4.2 Hardware Requirements

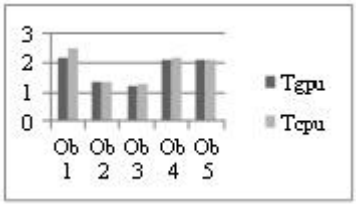
1. Switch
2. Server: Intel Xenon Processor ES 2603, 4 Cores
3. GPU : Nvidia GeForce GTX 680,1563 CUDA Cores, 1006 MHz Base Clock

5. RESULT ANALYSIS

During initial testing phase for converting approximately a 2MB avi video file, the following observation on CPU and GPU based conversion has been taken.

Table 1: Observation Table

Sr. No.	T_{GPU}(sec)	T_{CPU}(sec)
1	2.1368	2.5246
2	1.3246	1.3629
3	1.2468	1.2893
4	2.1234	2.1645
5	2.0645	2.1032



Graph 1: Observation Graph

Based on the observation made on Table 1, Graph 1 has been generated. From the above Graph 1, we can clearly see that time taken by CPU to convert a video is more than the time taken by the GPU. Five observations has been taken and based on that the average time has been calculated.

The average GPU time is 1.77922sec and average CPU time is 1.8889sec. So the conversion rate is increased by 0.010968%.

6. RESULT ANALYSIS

In this paper, based on the related work and the system developed, we can say that GPU can be used in Cloud environment. Various applications and methods are being implemented in GPU based cloud which shows improved performance and efficiency when executed in parallel as compared with CPU serial execution.

Based on the experimental observation a 0.010968% of conversion rate has been achieved. Further optimization of code can be considered as a future scope for this system.

REFERENCES

[1] G. Taori, A. Taole, S.Kothawade, S. Srivastava, "GPU Accelerated Video Conversion in the Cloud", in *2015 International Conference on Pervasive Computing(ICPC)*, Pune, 2015, pp. 1-5.

[2] M.D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research " , in *Internet Computing, IEEE*, vol.13 , Issue: 5, 2009, pp. 10-13

[3] Atsushi Kawai, Minoru Oikawa, Kentaro Nomura, Kenji Yasuoka, Kazuyuki Yoshikawa, Tetsu Narumi, "DS CUDA: a Middleware to Use Many GPUs in the Cloud Environment", in *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*, Salt Lake City, UT, 8, 2012, pp. 1207-1214

[4] Bingsheng He, Jianlong Zhong "Towards GPU Accelerated Large-Scale Graph Processing in the Cloud", in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, vol. 1, Bristol, 8, 2013, pp. 9-16

[5] Ryan Shea, Jiangchuan Liu, "GPU Pass-Through Performance for Cloud Gaming: Experiments and Analysis", in *2013 12th Annual Workshop on Network and Systems Support for Games(NetGames)*, Denver, CO, USA, 6, 2013, pp. 1-6

[6] Kiatchumpol Suttisirikul, Putchong Uthayopas, "Accelerating the Cloud Backup using GPU based Data Deduplication", in *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, Singapore, 4, 2012, pp. 766-769.

[7] Damien Vintache, Bernard Humbert, David Brasse, "Iterative reconstruction for transmission tomography on GPU using Nvidia CUDA", in *Tsinghua Science and Technology*, vol. 15, issue 1, 2012, pp. 11-16.

[8] Zhiyi Yang, Yating Zhu, Yong Pu,"Parallel Image Processing Based on CUDA", in *2008 International Conference on Computer Science and Software Engineering*, vol. 3, Wuhan, Hubei, 4,2008, pp. 198-201.

[9] Matthias Leinweber, Lars Baumg¨artner, Marco Mernberger, Thomas Fober, Eyke H¨ullermeier, Gerhard Klebe, Bernd Freisleben,"GPU-based Cloud Computing for Comparing the Structure of Protein Binding Sites", in *2012 6th IEEE International Conference on Digital Ecosystems Technologies(DEST)*, Campione dItalia, 6, 2012, pp. 1-6.

[10] Sheng-Ta Lee, Chun-Yuan Lin, Che Lun Hung, Hsuan Ying Huang," Using Frequency Distance Filteration for Reducing Database Search Workload on GPU-Based Cloud Service", in *2012 IEEE 4th International Conference on Cloud Computing Technology and Science*, Taipei, 6, 2012, pp. 735-740.

[11] Khaled M. Diab, M. Mustafa Rafique, Mohamed Hefeeda," Dynamic Sharing of GPUs in Cloud Systems", in *2013 IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum*, Cambridge, MA, 8, 2013, pp. 947-954.

BIOGRAPHIES

Sandip M. Walunj
Working as Asst. Professor in the department of Computer Engineering at the Sandip Institute of Technology and Research Centre. His area of research includes Parallel Processing and CUDA.

Akash Talole
Student at the Department of Computer Engineering at Sandip Institute of Technology and Research Centre. His interests include parallel processing and cloud computing.

Gaurav Taori
Student at the Department of Computer Engineering at Sandip Institute of Technology and Research Centre. His interests include cloud computing and business management.

Sachin Kothawade
Student at the Department of Computer Engineering at Sandip Institute of Technology and Research Centre. His interests include parallel processing and android app. Development.

