A TRANSACTION PAPER ON MULTIMEDIA DATA COMPRESSION IN DISTRIBUTION SYSTEMS VIA SVD AND HADOOP

Nilesh M. Deshmukh

CSE Department, MSSCET, BAMU, Maharashtra, (India)

ABSTRACT

The tremendous growth of Internet technologies such as social networking services (SNSs) allows users to use and disseminate multimedia data like images and videos. In this paper, we have developed the Split and Merge architecture for high performance video processing, a generalization of the MapReduce paradigm that rationalizes the use of resources by exploring on demand computing. To illustrate the approach, we discuss an imple-mentation of the Split and Merge architecture. This reduces video encoding times to fixed duration, independently of the input size of the video file, by using dynamic resource provisioning. The traditional approach to transcoding data requires specific and costly hardware because of the high-capacity and high definition features of video data. So as to overcome the problem of performance, we have developed a new system which is the fusion of Hadoop and data processing. We are adopting Map Reduce for distributed and parallel processing of data. Image compression techniques are the most concerned topics in today's technological developments. Singular Value Decomposition (SVD) is one such image compression technique. This SVD performs its operations on matrices. In this paper, we have applied SVD algorithm on images for compression.

Keywords* : *Hadoop Distributed File System (HDFS); Singular Value Decomposition (SVD)

I. INTRODUCTION

In Data compression, we deal with large amount of data requiring large amount of processing, storage and communication resources. In this project, we have designed and implemented the compression of an data which is based on MapReduce strategy and Hadoop to solve the above mentioned problems.

The intention to use MapReduce framework here is that it provides a specific pro-gramming model and a run time system for processing and creating large data sets amenable to various real-world task. MapReduce programming model is executed in two main steps called as mapping and reducing. Mapping and reducing are defined by mapper and reducer where as mapping step receives input data sets and feeds each data element to mapper in the form of key and value pairs. In reducing step, all outputs from the mapper are processed and final result is generated by reducer using the merging process. HDFS (Hadoop Distributed File System) is used by Hadoop application which creates the replicas of blocks, distributes them on computed nodes throughout a cluster. This implementation is divided into two parts:

In first part we store the large amount of data into HDFS and second part deals with processing of data which is stored in HDFS using MapReduce and JAI (Java Advanced imaging) for compression data into target formats. In order to support these methods. Imple-mentation of this work starts with taking multimedia data as input by using record Reader method of the class inputFormat. InputFormat transforms the data into sets of keys and values and further they are passed to mapper. Mapper pro-cesses video data using JAI and compression module which compression the video data into specific formats suitable for smart-phones, pads and personal computers. After completing video data compression, mapper send the result to outputFormat then RecordWriter method of outputFormat class writes the result as a file to HDFS.

II. DATA COMPRESSION VIA SVD

A. Data Matrix

Consider, for example, that measurements from *m* different meters, taken over *t* time instants, need to be transmitted through the communications network of a smart distribution system to serve as inputs for a given application. Let this set of measurements be put in the form of a matrix **X**, with each row of **X** containing the measurements taken from a given meter at each time instant.

.B. Data Compression

Data matrix **X** can be factorized into three matrices by applying the SVD. In order to simplify
the notation, matrix **VT** is hereafter written as just **V**

$\mathbf{X}(\mathbf{m}\times\mathbf{t}) = \mathbf{U}(\mathbf{m}\times\mathbf{m})\ (\mathbf{m}\times\mathbf{t})\mathbf{V}(\mathbf{t}\times\mathbf{t})$

where diagonal matrix contains the SVs of **X**, ordered from the highest to the lowest. Data compression can be achieved by taking advantage of the fact that many matrices occurring in practice do exhibit some kind of structure that leads to only a few SVs actually being non-negligible. In such cases, good approximation of
matrix **X** can be obtained by keeping only the SVs found to be significant in matrix . Assume that *r* SVs are to be retained in and let matrix **XR** denotes the approximated matrix **X**.

C. Compression Ratio

The extent of compression achieved by a coding scheme can be measured by a CR. The term CR has been defined in several ways in the literature. In many contexts, the CR is computed by dividing the size of the original data by the size of the compressed data. A CR = 4, for example, means that the data has been compressed with the ratio 4:1. Alternatively, it can be said that the volume of the compressed data is 25% of the original data. In this paper, the CR is computed, which expresses the ratio between the total number of elements in the original matrix **X** (measurements) and the total number of elements in the submatrices that are needed to compute matrix **XR**

CR

$= \ m \times t$

International Journal of Advance Research in Science and Engineering
Vol. No. 5, Issue No. 08 , August 2016
www.ijarse.com
IJARSE
ISSN 2319 - 8354

$(m + t + 1) \times r$

D. Loss of Information

As discussed, lossy compression methods can be very effective for data compression, but this comes with a cost, which is the loss of information that will not be retrieved when the original data is reconstructed. Then, data compression should be carried out in a way that a good tradeoff between the CR and loss of information is achieved. In other words, data compression should not result in loss of information that renders the reconstructed matrix of limited use to the applications that would employ it as input data.

In this paper, the loss of information is measured in terms of the mean absolute error (MAE) and the mean percentage error (MPE) observed when comparing the reconstructed data matrix with the original one.

II. FIGURES AND TABLES

Fig. Use-case diagram

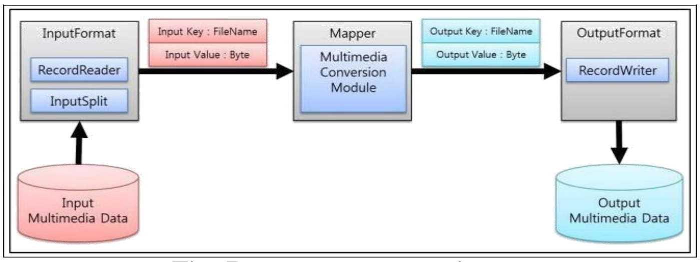

Fig. Base paper conversion

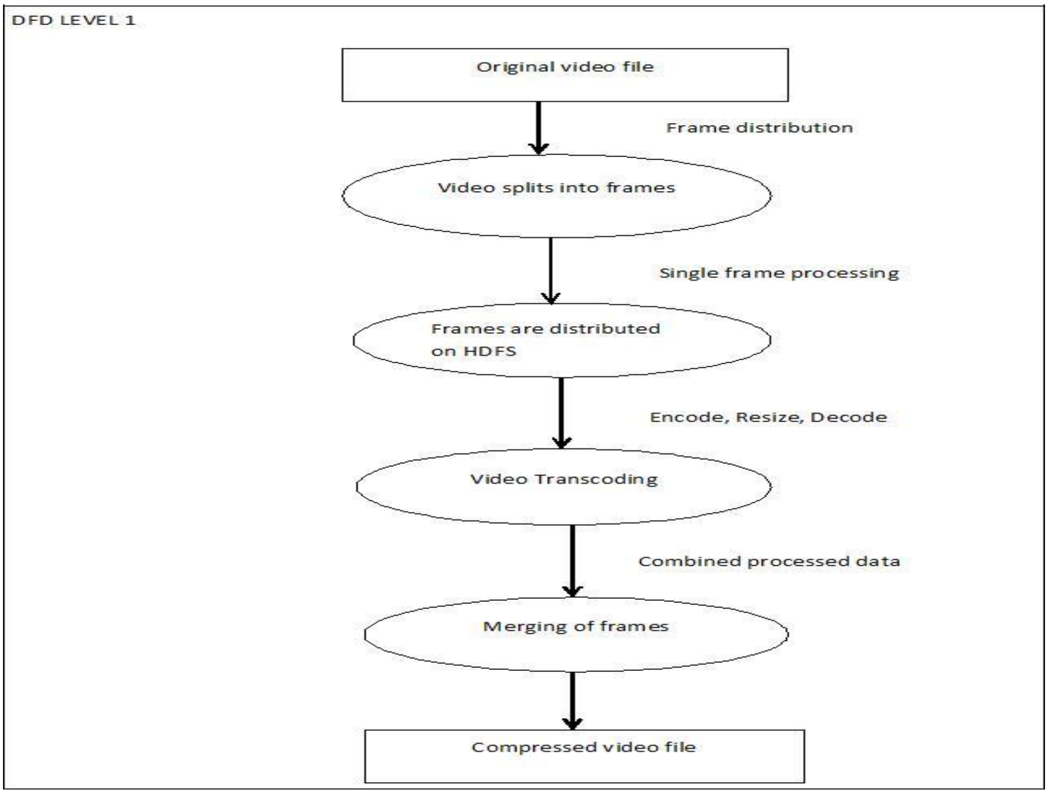

Fig. DFD Level 1

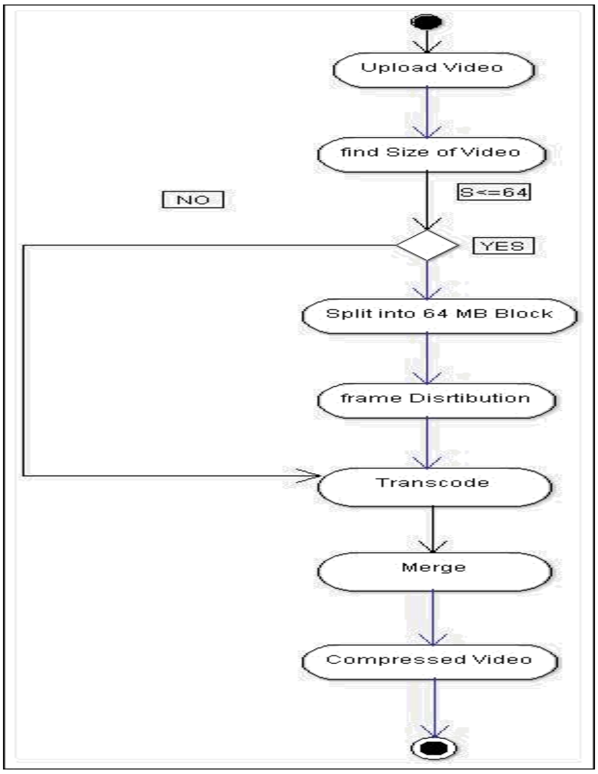

Fig. Activity Diagram

Table – Risk Table

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Failure of database	Low	Low	High	High
2	Failure of node	Low	Low	High	High
3	Module integration	High	High	High	High
4	Order of frame not maintain	Low	Medium	High	Low
5	Quality gradually decrease	Medium	Medium	High	High

Table – Image Compression Results for SVD

Sr.no	Time	MSE	PSNR	Original Size(kb)	Compressed Size(kb)	Compression Ratio(%)
1 (Chrysanthemum.jpg)	198.0 ms	228.0	24.55	858.78	18.39	46.67
2 (Desert.jpg)	141.0 ms	288.0	23.53	826.11	16.87	48.96
3 (Hydrangeas.jpg)	129.0 ms	175.0	25.70	581.33	16.84	34.51
4 (Jellyfish.jpg)	128.0 ms	206.0	24.99	757.52	16.04	47.20
5 (koala.jpg)	141.0 ms	282.0	23.62	762.53	23.01	33.13

Table – Results For Video Compression

Sr No	Original Size(kb)	Compressed Size(kb)	Time
1.	1024	147	44 sec
2	262	231	17 ec

III. CONCLUSION

In this project, we tend to compress any high definition video uploaded by a user. The video data needs to be stored in a database, so to reduce the storage problem we have used Hadoop Distributed File System(HDFS). This project is useful for the companies working with huge amount of multimedia data. This project will be helpful in saving buffering time and providing satisfactory results to the user while watching the video. We use Parallel Distributed environment by establishing the nodes on our laptops. By using this environment, the data is distributed and is processed in parallel manner thus resulting in achieving the end state(compressed video) in less time. The module is based on Hadoop HDFS and the MapReduce framework for distributed parallel processing of large-scale video data. We redesigned and implemented Input Format and Output Format in the MapReduce framework for image data and used SVD. We used the JAI library for converting the video format and compressing the videos.

IV. ACKNOWLEDGEMENTS

It is pleasant Endeavour to present Transaction paper report on **"Multimedia Data Compression in Distribution Systems via SVD And Hadoop"** I take this opportunity to express my gratitude towards my guide **Prof.G.P.Chakote** for his constant encouragement and guidance. He is a constant source of motivation and inspiration. Without his efforts the project could not have taken final phase.

I would also like to thank to **Prof.Ruhi Sodani** and **Dr.C.M.Sedani** for their cordial support who have co-operated and provided valuable information to complete this Project.

I would like to extend my gratitude to all my friends, who have extended their support and helped me to complete Project Report. Words fall short to express my deep sense of gratitude towards them all, who have directly or indirectly helped in making this project.

REFERENCES

[1] Book Pressman, Tom White, "Hadoop: The Definitive Guide" 4th Edition, year = 2015, isbn = 978-1-491-90163-2, 1491901632, 978- 1491901632, publisher = McGraw-Hill, Inc., address = New York, NY, USA

[2] A. Ghassemi, S. Bavarian, and L. Lampe, "Cognitive radio for smart grid communications," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, Gaithersburg, MD, USA, 2010, pp. 297–302.

[3] G. T. Heydt, "The next generation of power distribution systems," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 225–335, Dec. 2010.

[4] G. Artale *et al.*, "Medium voltage smart grid: Experimental analysis of secondary substation narrow band power line communication," *IEEE Trans. Instrum. Meas.*, vol. 62, no. 9, pp. 2391–2398, Sep. 2013.

[5] K. Shvachko, H. Kuang, S. Radia and R. Chansler, The Hadoop distributed file system, in Proc. 2846 Kim et al.: A Hadoop-based Multimedia Transcoding System for Processing Social Media of 2010 IEEE 26th Symposium on Mass Storage Sys-tems and Technologies, pp.1-10, May. 2010.

[6] A. Ghassemi, S. Bavarian, and L. Lampe, "Cognitive radio for smart grid communications," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, Gaithersburg, MD, USA, 2010, pp. 297–302.

[7] Ishfaq Ahmad, Xiaohui Wei, Yu Sun, Student Member, and Ya-Qin Zhang "Video Transcoding: An Overview of Various Techniques and Research Issues" (2005, IEEE).

[8] Myoungjin Kim1, Seungho Han1, Yun Cui1, Hanku Lee1,* and Changsung Jeong2. "A Hadoop-based Multimedia Transcoding System for Processing Social Media in the PaaS Platform of SMCCSE "