e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

AN APPROACH FOR FAST AND PARALLEL VIDEO PROCESSING ON APACHE HADOOP CLUSTERS

R. Jagadish Vijay*1

*1Assistant Professor, Department Of Electrical And Electronics Engineering, Amrita College Of Engineering And Technology, Erachakulam, Tamil Nādu, India.

DOI : https://www.doi.org/10.56726/IRJMETS33990

ABSTRACT

In this paper, a method for quick and concurrent video processing on Map Reduce-based clusters like Apache Hadoop is proposed. The technique can handle massive amounts of video data and processing time can be drastically decreased by using clusters. The mechanism for transferring conventional video processing algorithms created for a single computer to the suggested system is revealed, along with other technical aspects of doing video analysis on clusters. Algorithms for face detection, motion detection, and tracking have been put into practice on clusters as case studies. Performance tests on a six-machine Apache Hadoop cluster demonstrate that the system is able to lower the two implemented algorithms' operating times to under 25% of those of a single computer. The applications of the system include smart city video surveillance, services provided by video sites and satellite image processing

Keywords: Hadoop, Map Reduce, Video Processing, Face Detection, Motion Detection And Tracking.

I. INTRODUCTION

The Map Reduce paper [11], written by Jeffrey Dean and Sanjay Ghemawat, established the groundwork for data processing on Map Reduce-based clusters. Map Reduce has been used for a long time to handle massive volumes of data on inexpensive processors. Video processing, which is by nature data-intensive, can benefit from Map Reduce to speed things up. The value of combining video processing with Map Reduce has already been recognized by certain academics. On Map Reduce-based clusters, the architecture [7] and particular applications like video transcoding [8] are described. The notion of using Map Reduce-based clusters for video processing is clearly demonstrated by Rainer Schmidt et al. [10], who also talk about multimedia processing using Hadoop. Several technical specifics are not provided, though. In this paper a different architecture is proposed, which is much more programming friendly and is able to perform complicated video processing with technique details revealed.

II. SYSTEM ARCHITECTURE AND IMPLEMENTATION

Here, the challenge is how to speed up challenging video processing while utilizing Apache Hadoop clusters. A software foundation for data-intensive applications is provided by systems based on map-reduce, such as Apache Hadoop. However, because of the following two reasons, it is not handy for video processing:

1) Changes have to be made to make common video processing libraries available for the system since they are originally designed for local applications.

2) Changes have to be made for video processing algorithms to take advantage of Map Reduce Programming model.

The proposed system is to embed those changes into Apache Hadoop system, making it programming-friendly for complicated video processing.

System Components

The Hadoop system that is being suggested is made up of the Apache Hadoop project itself, its child project Fuse-DFS, the video processing libraries OpenCV and FFMPEG, as well as JavaCV, an open-source project hosted by Google Code. We will briefly introduce those components and then focus on the overview of the system.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

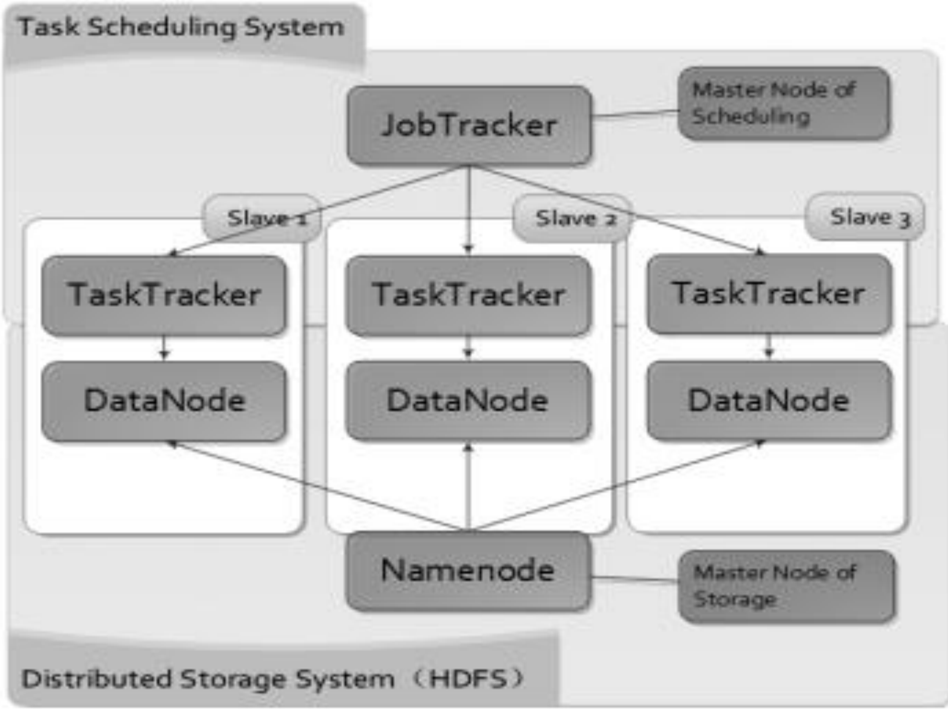

Figure 1: Apache Hadoop System Architecture

Apache Hadoop System Architecture

As shown in Figure 1, Apache Hadoop may be broken down into two sub-systems: the task scheduling system and the distributed storage system (Hadoop's Distributed File System, HDFS).

The master of Task Trackers, Job Tracker distributes Map Reduce work around the cluster. The Data nodes that control the distributed storage system are led by Name node. A Task Tracker and a Data node are deployed with each slave node. For performance, Map Reduce tasks should be dispatched to slaves near the data.

Fuse-DFS project

Fuse-DFS is an Apache Hadoop subsidiary project. In order for the millions of programming libraries created for local file systems to benefit from Hadoop, it offers an interface to fill the gap between HFDS and the local file system. Be aware that this project is dependent on the Fuse Hadoop subproject.

III. SYSTEM OVERVIEW

With components introduced in previous section, we can build the Hadoop system for video processing. Figure 2 briefly demonstrates the system structure.

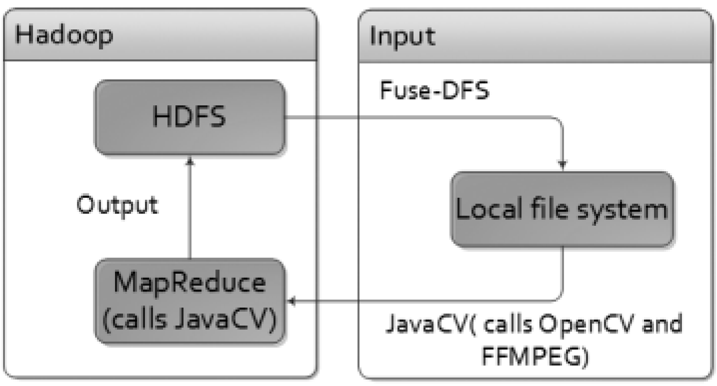

Figure 2: System Overview

In Figure 2, we can see that:

1) HDFS offers distributed storage service for video data.

2) Fuse-DFS mounts the distributed file system to local file system.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

3) JavaCV ports two video processing libraries, OpenCV and FFMPEG, to Java programming language. So they are available for the whole video processing procedure.

4) Map Reduce programming model of Hadoop processes video data concurrently.

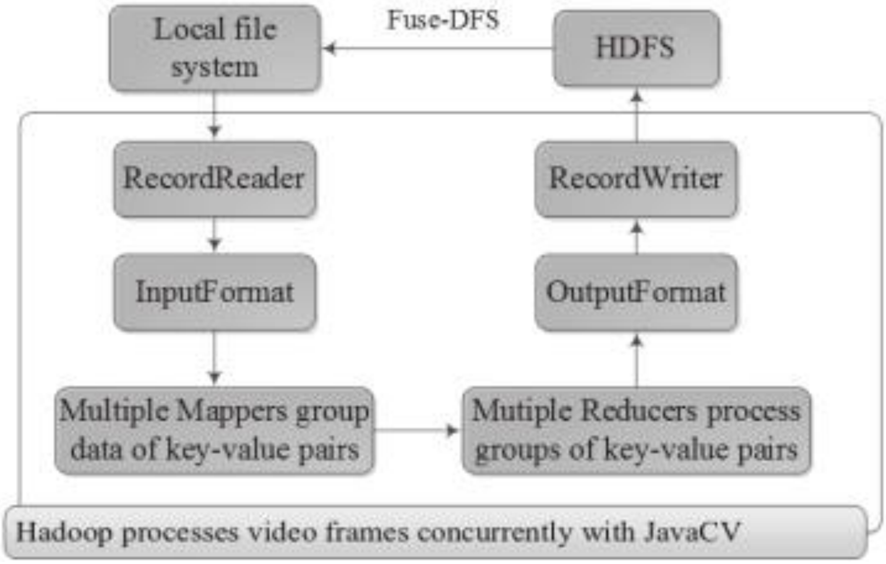

Figure 3: Programming Model of the proposed Hadoop System

The key idea of Map Reduce programming model is to encapsulate data into key-values pairs so that Mappers and Reducers can concurrently process them in parallel streamlines, which dramatically increases the throughput of the system. Figure 3 demonstrates the programming model of the proposed Hadoop Cloud system for video processing.

It can be seen from Figure 3 that Fuse-DFS mounts HDFS to local file system and make video data stored on HDFS available to JavaCV. JavaCV, which inherits powerful video analysis ability from OpenCV and FFMPEG, makes the video libraries available to the whole processing procedure from video IO to MapReduce.

IV. EXPERIMENTS

Face Detection

Figure 4& 5 illustrate the performance of face detection. For the face detection program running on Hadoop, it can be concluded that the running time.

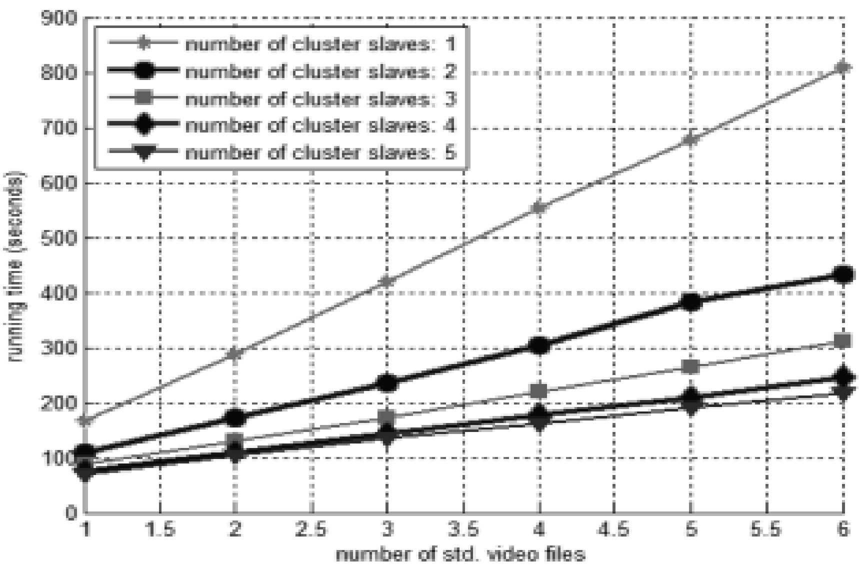

Figure 4: Running time with different number of std. input videos

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:03/March-2023 Impact Factor- 7.868 www.irjmets.com

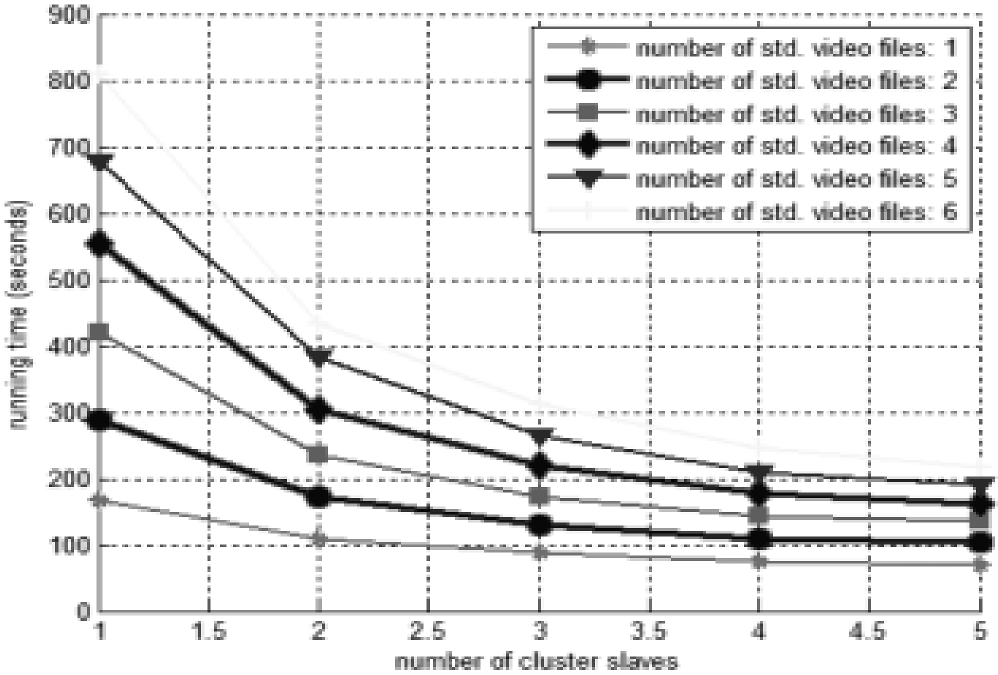

Figure 5: Running time with different number of cluster slaves for face detection

1) Increases approximately linearly with the size of input video files growing. This is natural since face detection is a simple repeated work on frames.

2) Decreases significantly with the increase of number of cluster slaves. The more time the task consumes on a single computer, the more time it saves to do the task with Hadoop cluster. A cluster with five slaves can reduce the execution time to less than 25% of that of a single computer. And we can infer that the running time can be further reduced with more nodes added.

V. CONCLUSION

In this paper, a method for quick and concurrent video processing on Map Reduce-based clusters like Apache Hadoop is proposed. The technique can handle large-scale video data by using clusters, and the running time may be greatly decreased. The system design, the way algorithms are implemented on the system, and experiments to show the system's effectiveness are all introduced in this paper. The solution that is being introduced is made up of the original Apache Hadoop, its child projects Fuse and Fuse-DFS, the well-known video processing libraries FFMPEG and OpenCV, as well as JavaCV, its Java interface project. As examples of how to build a video processing method on the system, we use the face detection algorithm and the motion detection and tracking algorithm. Six computers make up an Apache Hadoop cluster, and performance tests on this cluster have revealed that the system can function at less than 25% of the speed of a single computer.

VI. REFERENCES

[1] Apache Hadoop project description on Wikipedia, http://wiki.apache.org/hadoop/ProjectDescription, retrieved on December 28, 2012.
[2] Tom White, "Hadoop the Definitive Guide", O'Reilly | Yahoo! PRESS, 2009.
[3] YANG Fan, SHEN Qi-Wei, "Distributed Video Transcoding on Hadoop", Computer System Applications, 20(11):80-85, 2011.
[4] ZHU Yi-ruing, "Image Classification Based on Hadoop Platform", Journal of Southwest University of Science and Technology, 26(2):70-73, 2012.
[5] Mountable HDFS on Wikipedia, http://en.wikipedia.org/wiki/MountableHDFS, retrieved on January 20, 2013.
[6] JavaCV project page on google code, http://code.google.com/p/javacv, retrieved on August 20, 2012.

[7] R. Pereira, M. Azambuja, K. Breitman, M. Endler, "An Architecture for Distributed High Performance Video Processing in the Cloud", 2010 IEEE 3rd International Conference on Cloud Computing: 482-489, 2010.

[8] A. Garcia, H. Kalva, "Cloud transcoding for mobile video delivery", 2011 IEEE International Conference on Consumer Electronics (ICCE):379-380, 2011.

[9] Dejan Kovachev, Yiwei Cao, and Ralf Klamma, "Mobile Multimedia Cloud Computing and the Web", 2011 Workshop on Multimedia on the Web, 2011.

[10] Rainer Schmidt and Matthias Rella, "An Approach for Processing Large and Non-uniform Media Objects on MapReduce-Based Clusters", Digital Libraries: For Cultural Heritage, Knowledge Dissemination, and Future Creation, Springer Berlin Heidelberg, 2011.

[11] Jeffrey Dean and Sanjay Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", OSDI, 2004.