ISSN:0975-9646

Large Scale Image Processing Using Distributed and Parallel Architecture

Helly M. Patel[#1] , Krunal Panchal[#2] , Prashant Chauhan[+3] , M. B. Potdar [*4]
[#]*L.J Institute of Engineering and Technology, Ahmedabad, Gujarat, India*
[+]*BISAG, Gandhinagar, Gujarat, India*
[*] *BISAG, Gandhinagar, Gujarat, India*

***Abstract* ---Enormous amount of images are uploaded and used via internet and this ratio is still drastically increasing .So there is instant need to handle this data and customize and filter it as per user and application requirement. This paper describe some parallel and distributed processing techniques like Hadoop, HIPI, Map reduce, CUDA, MPI to process massive database. Mapreduce based large-scale images processing, which exhibit high reliability and scalability in distributed and parallel environment. As highly efficient parallel data processing all these methods are used to analyze and process massive image database. In today's highly digitalized world large volume of multimedia data need to efficiently store, process and analyze This paper shows usefulness of parallel and distributed techniques.**

***Keywords*---Image Processing, Parallel and distributed processing, Mapreduce, Hadoop, HIPI, CUDA, MPI**

I. INTRODUCTION

Large amount of image data is generated in present scenario due to social media, satellite image, surveillance camera, and medical Image data. So at a same time there is a need to develop techniques and algorithms to analyze this data with high efficiency and in timely manner. Image processing consists of manipulating the image in order to obtain a desired image result for different purposes including visualization, image retrieval, and image recognition[10]. Many Image processing algorithm and techniques are developed for high level and efficient image processing but most of them are resource and time intensive Traditional approach to analyze multimedia data requires specific and expensive hardware because of the high-capacity and high definition features of multimedia data and imposes a considerable burden on the computing infrastructure as the amount of data increases[2]. Especially in the case of satellite image processing high computational, processing resources and network bandwidth is required to process chunks of images on distributed nodes. .Images need to process in distributed and parallel environment to achieve speed up efficiency and faster execution. Image dataset is divided into different parts and then processed in parallel manner on different processors or through distributed clusters .In distributed system images are processed by different nodes at different geographical locations. As distributed and parallel processing provide high Reliability, elastic scalability and fault tolerance for incremental and real time dataset like satellite data , it served as efficient candidate to process such dataset.

II. RELATED WORK

Hossein Kardan Moghaddam et al. proposed MapReduce as a distributed data processing model using open source Hadoop framework for manipulating large volume of data [9]. Muneto Yamamoto et al. suggested methods parallel image database processing with mapreduce and Hadoop streaming [8]. Euiseong Seo et al. proposed an extensible video processing framework in Apache Hadoop to parallelize video processing tasks in a cloud environment with the use of FFmpeg for a video coder, and OpenCV for an image processing engine [6]. Lidong Chen et al. suggested method for fast and parallel video processing on MapReduce-based clusters through Fuse-DFS, FFMPEG, OpenCV and JavaCV [7]. Roberto Giachetta developed a geospatial data processing framework designed to enable the management and processing of spatial and remote sensing data in distributed environment [4]. Hong Zhang et.al. developed scalable and distributed geographic information system, called Dart, based on Hadoop and HBase which provides a hybrid table schema to store spatial data in HBase[1] Vaibhav Nirgun et. Al. proposed Hadoop HDFS and MapReduce for distributed parallel processing of image database and JAI library for converting the image database into target format and resizing the images also convert the resizing images into grey scale format[2]. Stefan Lee et.al. suggested Map-Collective model for Large Scale Image Classification using High Performance Clustering[3].

III. PARALLEL APPROACHES FOR DATA PROCESSING

A. MAP REDUCE

Map reduce is a framework for distributed parallel processing of large image database [8]..Map reduce model is having many different variation with different technology and framework .Google ,Apache Hadoop ,HIPI, Microsoft SCOPE, Apache Pig, and Apache Hive all these have their own customized map reduce implementation.

1) Hadoop Mapreduce: System Architecture

Hadoop is an open source, distributed, scalable java based implementation which follows Google's MapReduce concept [9.] Hadoop is framework which is having its own distributed file storage system which is Hadoop Distributed File System (HDFS) and its own computational paradigm known as Map reduce[12].

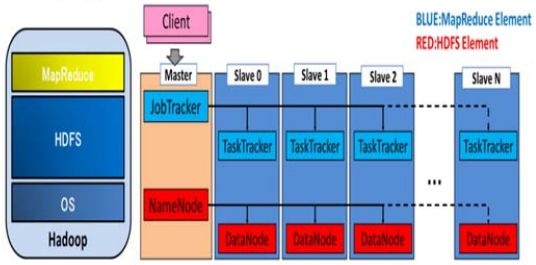

Fig.1 Hadoop MapReduce Paradigm [8]

While processing Data through Hadoop Input and output is always given through HDFS Mapreduce is having two main elements namely JobTracker and TaskTracker and Two functions namely Map and Reduce. HDFS is having 2 main elements namely Name node and Data node

A) JobTracker manage resources of distributed system and manage job scheduling [8].
B) TaskTracker accepts task and returns the results after executing tasks received by JobTracker.
C) Name node is a master server that manages the namespace and access to files by client's Name and executes file operations, such as opening, closing, and renaming files and directories. It also determines the mapping of blocks to Data Nodes[8]
D) DataNodes manage the storage that is attached to the nodes on which they run and perform block creation, deletion, and replication[8] .It is a place where execution of task take place.
E) SecondaryNameNode is a helper to the primary NameNode responsible for supporting periodic checkpoints of the HDFS metadata[8].It is especially useful in case of primary name node failure
F) Map task take multiple input key- value pairs <k,v> and generate multiple <k',v'> intermediate pairs.

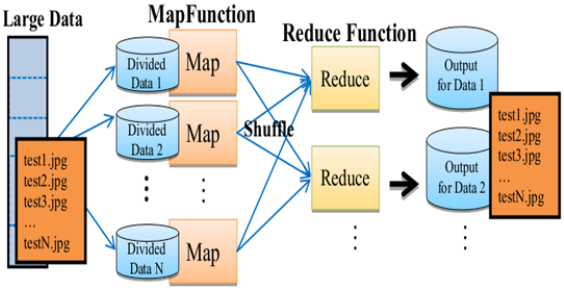

Fig 2 working of Map and reduce phase [8]

G) Reduce phase take list of input<k', list v'> and give final summarized output.
H) After Map task shuffle task is performed on intermediate values to efficiently aggregate different pairs and to save network bandwidth.

2) HADOOP IMAGE PROCESSING INTERFACE (HIPI)

HIPI is an image processing library designed to be used with the Apache Hadoop Mapreduce parallel programming framework [5]. HIPI facilitates efficient and high-throughput image processing with MapReduce style parallel programs typically executed on a cluster [11]. It is flexible enough to withstand continual changes and improvements within Hadoop's Mapreduce system. The goal of HIPI is to create a tool that will make development of large-scale image processing and vision projects extremely accessible [18].

Primary objective of HIPI are as below:

1) Provide an open source framework over Hadoop MapReduce for developing large-scale image applications [5].

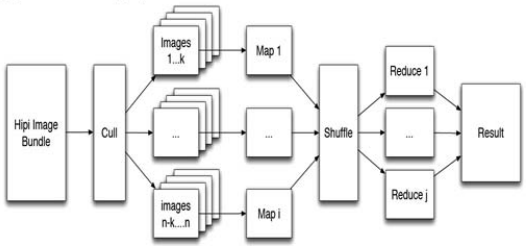

Fig 3 Organization of Mapreduce in HIPI

2) Provide the ability to flexibly store images in various Hadoop file formats [11].
3) Provide interoperability between various image processing libraries [5].
4) Store images efficiently for use in MapReduce applications and simple filtering of a set of images [18].
5) HIPI will set up applications so that they are highly parallelized and balanced so that users do not have to worry about such details [18].

Working of HIPI in MapReduce environment is as follow:

1) Input to the HIPI program is given in the form of HIPI Image Bundle (HIB).HIB is collection of images in variety of file format which is stored as a single file on the HDFS [18].
2) HIB is given to culling phase, which is new in HIPI. Main goal of culling step is to filter the images in a HIB based on a variety of user-defined conditions like spatial resolution or criteria related to the image metadata. This functionality is achieved through the CullMapper class [18].
3) Images survive from cull step are given to map function to generate intermediate key value pairs [18].
4) Mapping output is shuffled to minimize network bandwidth usage and pre aggregate key value pairs [18].
5) Reduce phase will generate summarized data in the form of one key with multiple values pair [18].

B. MPI

MPI (Message passing Interface) is message passing library designed for parallel programming through distributed shared memory .Basic communication is handled through send() and receive() primitive for passing message. It provide standard message passing operation with synchronous and asynchronous variants [17]. It provide portable efficient and fault tolerant environment for parallel programming on shared memory. It provide powerful and general way of expressing parallelism.

C. CUDA

Compute Unified Device Architecture (CUDA) is an architecture and programming model that allows leveraging the high compute-intensive processing power of the Graphical Processing Units (GPUs) to perform general, non-graphical tasks in a massively parallel manner [16].It is a programming environment based on GPU in which CPU is host and GPU is coprocessor [13]. It is composed of a number of Streamlined Multi-Processors (SMs) each of which is composed of a number of processing cores. Each SM can execute the same function in SIMT (Single Instruction Multiple Threads) fashion using a limited number of threads organized as a thread block [15]. Each SM (thread block) has a shared memory and a set of Registers, accessible to all the threads of that SM [15].

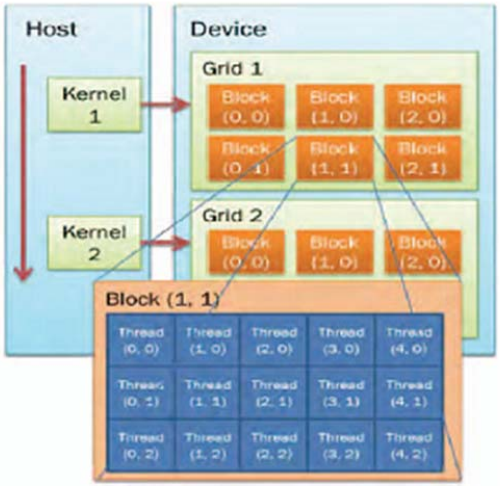

Fig 4 Threaded hierarchy of CUDA [14]

IV IMAGE PROCESSING ALGORITHMS

For the purpose of performance evaluation we had implemented two very commonly used image processing algorithm.

A) Canny edge detection algorithm

Description and extraction of features from image is an important task useful for a wide range of application fields such as object recognition, image segmentation, data compression, land-water border etc. Edges in an image are signified by a significant image intensity change which represents important object features and boundaries between objects in an image[12]. This multistep algorithm is considered as a standard and optimal detector among all edge detector algorithm

Three main objective of algorithm is

1) Good edge detection by maximizing the signal -to -noise ratio meaning the method should detect edges to the maximum possibility but with low probability of detecting edges falsely.
2) The second criterion is that detected edges should be as close as possible to the real edges.
3) Minimal number of edges should be detected more than once.

We had implemented this algorithm in Ubuntu 15.04 with Hadoop 2.4.0 and Java version 1.7.0_79.Following results show that threshold value plays an important role in edge detection. Two parameter –high and low threshold is supplied to this algorithm. It is observed by giving different high and low value that by minimizing value of low threshold we can detect more no of edges and higher threshold value will detect less edges

B) Image Segmentation using K means Clustering

The k-means algorithm is an unsupervised clustering algorithm that classifies input data points into multiple classes based on their Euclidean distance from each other. The algorithm assumes that the data features form a vector space and tries to find natural clustering within it [6].It divides image into K no of clusters. We had implemented both continuous and Iterative K means algorithm with different values supplied for K and will get corresponding RGB Image and its execution time in ms(mille second). Results show that execution of continuous algorithm take more time than iterative algorithm.

Fig 5(a) Original Image

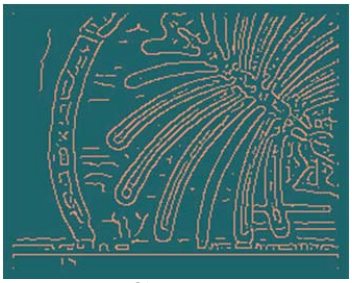

Fig 5(b) H.T -1, L.T -0.5

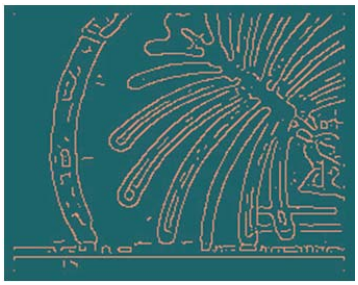

Fig 5(c) H.T -3, L.T -4.5

Fig 5.Canny Edge Detection with different High and Low Threshold Values. For Normal Image.(H.T – High Threshold , L.T –Low Threshold)

Fig 6(a) Original Image

Fig 6(b) H.T -2, L.T -2.5

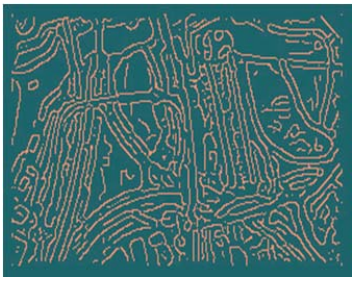

Fig 6(c) H.T -1, L.T -0.5

Fig 6(d) Original Image

Fig 6 (e) H.T -1, L.T -0.5

Fig 6 (f) H.T -3, L.T -4.5

Fig 6.Canny Edge Detection with different High and Low Threshold Values For satellite Image.

Fig 7(a) Original Image

Fig 7(b) K = 3(C-791 ms, I-741 ms)

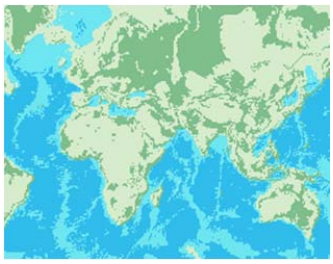

Fig 7(c) K = 4(C-665 ms, I - 653 ms)

Fig 7(d) Original Image

Fig 7(e) K = 3(C-635 ms, I - 616 ms)

Fig 7(f) K = 2(C-616 ms, I - 598 ms)

Fig 7.K means clustering with iterative and continuous algorithm running time (I-Iterative algorithm, C – Continuous algorithm)

V. CONCLUSION

Map reduce parallel programming model provide high scalability, reliability, fault tolerance in distributed environment. It provide sequential execution of map and reduce task. In this paper we discussed Hadoop and HIPI's map reduce implementation especially for image processing and computer graphics. Applications we had implemented canny edge detection algorithm and k means clustering algorithm. Despite of it efficiency, efficient input output methods to map reduce program is still an issue.

ACKNOWLEDGEMENT

This paper is carried out with the full support from Bhaskaracharya Institute for Space Applications and Geo-informatics and the director of Institute Mr. T. P. Singh. I am also thankful to all the members of the institute for supplying the precious data and resources.

REFERENCES

[1] Zhang, Hong, Zhibo Sun, Zixia Liu, Chen Xu, and Liqiang Wang, "*Dart: A Geographic Information System on Hadoop*." In Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on, pp. 90-97. IEEE, 2015.

[2] Barapatre, Mr Harish K., Mr Vaibhav Nirgun2 Mr Harish Jagtap, and Mr Sagar Ginde, "*Image Processing Using Mapreduce With Performance Analysis.*"

[3] Zhang, Bingjing, Judy Qiu, Stefan Lee, and David Crandall, "*Large-Scale Image Classification using High Performance Clustering*."

[4] Giachetta, Roberto, "*A framework for processing large scale geospatial and remote sensing data in MapReduce environment.*" ,Computers & Graphics (2015).

[5] Vemula, Sridhar, and Christopher Crick, "*Hadoop Image Processing Framework.*"

[6] Ryu, Chungmo, Daecheol Lee, Minwook Jang, Cheolgi Kim, and Euiseong Seo, "*Extensible video processing framework in apache hadoop*." ,In Cloud Computing Technology and Science (CloudCom), 2013 ,IEEE 5th International Conference on, vol. 2, pp. 305-310. IEEE, 2013.

[7] Tan, Hanlin, and Lidong Chen,"*An approach for fast and parallel video processing on Apache Hadoop clusters*." In Multimedia and Expo (ICME), 2014 IEEE International Conference on, pp. 1-6. IEEE, 2014.

[8] Yamamoto, Muneto, and Kunihiko Kaneko, "*Parallel image database processing with MapReduce and performance evaluation, in pseudo distributed mode*." International Journal of Electronic Commerce Studies 3, no. 2 (2013): 211-228.

[9] Banaei, Seyyed Mojtaba, and Hossein Kardan Mogha ddam, "*Hadoop and Its Role in Modern Image Processing*.", Open Journal of Marine Science 4, no. 04 (2014): 239.

[10] Asaduzzaman, Abu, Angel Martinez, and Aras Sepehri, "*A time-efficient image processing algorithm for multicore/manycore parallel computing*.",In SoutheastCon 2015, pp. 1-5. IEEE, 2015.

[11] Sweeney, Chris, Liu Liu, Sean Arietta, and Jason Lawrence, "*HIPI: a Hadoop image processing interface for image-based mapreduce tasks*.", Chris. University of Virginia (2011).

[12] Ermias Beyene Tesfamariam , Master of Science thesis "*Distributed Processing Of Large Remote Sensing Images Using Map Reduce*", February ,2011

[13] Wang, Zhanghu, Pin Lv, and Changwen Zheng, "*CUDA on Hadoop: A Mixed Computing Framework for Massive Data Processing.*" Foundations and Practical Applications of Cognitive Systems and Information Processing, Springer Berlin Heidelberg, 2014.

[14] KS, Manjunath Gowda, and Vinaykumar Gangadhar Hulyal, "*Parallel Image Processing from Cloud using CUDA and HADOOP Architecture: A Novel Approach.*" IJITR, 2015.

[15] Malakar, Ranajoy, and Naga Vydyanathan, "*A CUDA-enabled Hadoop cluster for fast distributed image processing.*" National Conference on Parallel Computing Technologies (PARCOMPTECH),. IEEE, 2013.0

[16] Reza, Motahar, Aman Sinha, Rajkumar Nag, and Prasant Mohanty, "*CUDA-enabled Hadoop cluster for Sparse Matrix Vector Multiplication*." 2nd International Conference on Recent Trends in Information Systems (ReTIS), IEEE, 2015.

[17] George Coulouris , Jean Dollimore , Tim Kindberg , Gordon Blair , *DISTRIBUTED SYSTEMS Concepts and Design* , Fifth Edition

[18] http://hipi.cs.virginia.edu/