Processing of Large Satellite Images using Hadoop Distributed Technology and Mapreduce : A Case of Edge Detection

Rupal Yadav
M.Tech in Computer Engineering
Department of Computer Science & Engineering
PES College of Engineering,Mandya
rupalyadav48@yahoo.com

Dr. M. C. Padma
Professor & Head of Dept.
Computer Science & Engineering
PES College of Engineering,Mandya
padmapes@gmail.com

Abstract— Now a day's amount of data continues to grow as more information becomes available. The Exponential growth of data and the increasing user's demand for real time satellite data has forced remote sensing service providers to deliver the required services. The processing of large amount of images is necessary when there are satellite images involved.This paper presents a distributed technology, mapreduce programming paradigm,which is based on Hadoop platform to process large-scale satellite images.The main aim of this hadoop concept is to take the advantage of high reliability and high scalability in the field of remote sensing as to achieve the purpose of fast processing of large satellite images.The Hadoop streaming technology is used in the model and the main operations are written on java as the mapper and reducer.The model has been implemented using virtual machines where the large number of images are delivered to the multicluster nodes for concurrent processing.This paper presents a MapReduce based processing of large satellite images using edge detection methods .Sobel, Laplacian, and Canny edge detection methods are implemented in this model.

Keywords- *Hadoop,Mapreduce,HDFS,Sobel,Laplacian,Canny.*

*****

I. INTRODUCTION

The remote sensing community has identified the challenge of processing large and complex satellite datasets to derive specified products,and to achieve the high-performance computing model in the field of remote sensing several effort has been made in previous few years. This study analyzes the recent advancements in distributed computing technologies in the MapReduce programming model and extends that for image processing of collected remote sensing images.This paper covers the topic of high performance computing in the field of remote sensing to address the computational requirement for processing of large remote sensing images. Due to increasing the size of large datasets it becomes difficult to access and process on a stand-alone,centralized processing server [3].Distributed computing infrastructures are suitable to store large-scale data like satellite images that have to be written only once and read frequently that makes it a low-cost supercomputing resources.MapReduce programming model, provides a paradigm for large-scale processing of satellite images on clusters of commodity computers [4].MapReduce is highly simplified distributed programming model that aim to process huge datasets in a parallel mode.

II. EXISTING SYSTEM

Current processing of large satellite images uses ordinary sequential ways to process this job.The program loads image after image, processing each image seperately before writing the newly processed image on a storage device.Most of these tools run on a single computer with a Windows operating system. Eventhough batch processing can be found in these single-processor,there will be problems with the processing of large satellite images on a single processor due to limited capabilities.Therefore, we are in need of a new distributed approach to work effectively on massed image data.

III. PROPOSED SYSTEM

The core aim of this project is to investigate how the processing of large satellite images can benefit from distributed computing environment through massive parallelization. In this study the MapReduce programming model is proposed as a framework for parallel processing of remote sensing images.This paper presents how the MapReduce programming model can be applied and tuned to process large satellite images [6].This paper implements various edge detection algorithms for enhancement and detection of edges from Landsat satellite images.

IV. MAPREDUCE PROGRAMMING AND HADOOP

The objective of the MapReduce framework is to extract the features of images that will be compared to the features of processing image operations. The main aim of this study is to test the feasibility of distributed processing of large satellite images using the MapReduce model to solve the problem of data bulkiness.The approach is to apply distributed technology to parallelize the process of the large satellite images by dividing the image into smaller images, process them by different processors in parallel way and merge to yield the final output [1].The assumption is that we have a large amount of satellite images to perform computation in low-cost commodity.The MapReduce programming model have also provides an interface to the application developer compared to the other models as only the map and reduce function are needed the whole distributed processing application besides the data input and output handling.But the problem is having control over issues such as where and when a mapper or reducer executes, which input chunks are processed by a specific node or mapper and which intermediate data is processed by a specific reducer.This brings a challenge to on how to optimize our algorithms to perform well especially if the distributed computing resource is heterogeneous in nature.

A.Mapreduce Programming Model

Map and reduce function both are defined in terms of key value(k,v)pairs.

1. map function: map(k1,v1)-->intermediate list(k2,v2)

2. reduce function: reduce(intermediate list(k2,v2))->list (v3)

Hadoop is designed to scale massively and solve the problem that involves analyzing large data(petabytes). hadoop is very fast for large jobs means if we want to process the large datasets,it works efficiently [2]. It makes it possible to run applications on multinode cluster of computer involving thousands of terabytes.

Hadoop in a nut shell is an operating system. It has two components:

(i)HDFS: HDFS is a file system designed for storing very large files in the system.
(ii)Map-Reduce: Application that works on the data stored in HDFS and act as resources scheduler.

V. SYSTEM ARCHITECTURE

The basic idea is to implement MapReduce to split the large input image into many small pieces and assigned small task to different devices or slave nodes.

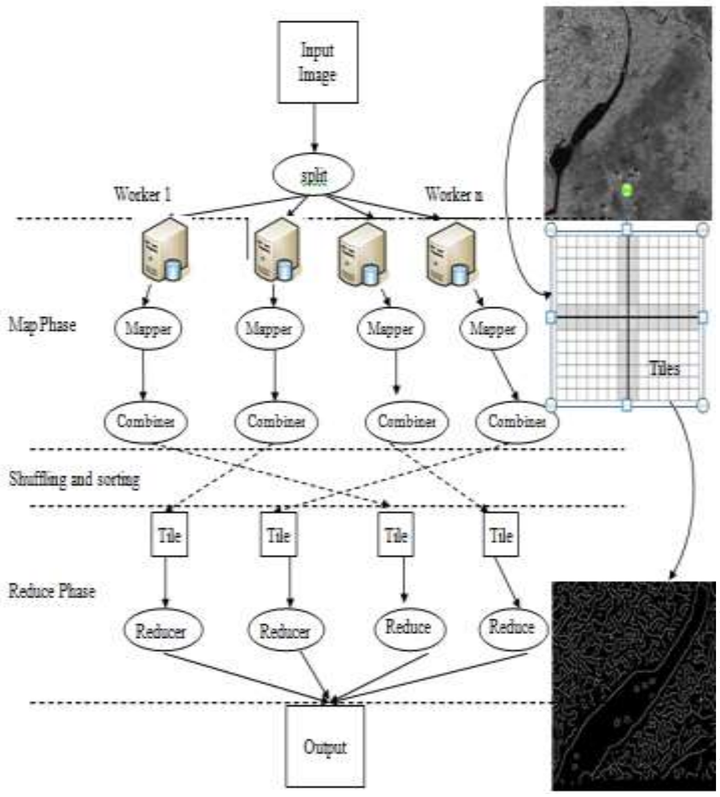

Fig1: System Architecture

Following are the working of the system:

1. Large no.of images stored in file system.
2. This bundle of images is fed to Hadoop Distributed File System.
3. Working with multinode cluster where 1 master node and 4 slave nodes.
4. Writing program for splitting the large image in RecordReader.
5. Writing a program for mapper and reducer whereas in mapper will write the program for splitting and streaming of images onto slave nodes and under reducer will write the code for shuffling and sorting of tiled images as well as reducer code.

VI. SOFTWARE IMPLEMENTATION

In this paper, we are using mapreduce programming model to implement the methodologies.Map job is basically used to read the large image file from the HDFS file and process it and finally write it back into HDFS file.Image files are considered as the key and content as the value pair.This scheme is applied to all image processing algorithms used in this paper.In order to specify the MapReduce ,the first thing is to consider when using Hadoop for image processing algorithms is how to specify its data types and function to read, write, and process binary image files in sequential Java programs. We can do that by changing the API of Hadoop to allow images in the BytesWritable class instead of the TextWritable class and stop the splitting of input files into chunks of data.

A. WholeFileInputFormat and Splitting

Hadoop can process many different types of data formats. We use a FileInputFormat class for implementation of InputFormat that uses files as their data source.
It provides two things : A place to define which files are included as the input to a job and the implementation for generating splits for the input files. FileInputFormat define two methods.
In the first method, the isSplitable() is override to return false and then ensure that input files are never split.
In the second method, the getRecordReader() returns a custom implementation of the RecordReader class .
FileInputFormat is responsible for creating the input splits and dividing those into records,the InputSplit is responsible for representing the data to be processed by one Mapper.

B. WholeFileRecordReader

The WholeFileRecordReader is responsible for taking a FileSplit and converting it into a single record. The record specifies the key and value for every record as well.WholeFileRecordReader has either processed it or not, so it maintains a Boolean called processed. If the file has not been processed when the nextKeyValue() method is called, then we open the file,create a byte array whose length is the length of file and use the IOUtils class to slurp the file into the byte array. Then we set the array on the BytesWritable instance that was passed into the next() method and return true to signal that a record has been read.

C. *Mapper*

The mapreduce applications are commonly written in java.It can also be written in other language [5].The mapper contains image processing algorithms means whatever methodologies is applied to every input image.large Satellite Input images are represented by the key–value pair to generate an intermediate key-value pair and store it back into the HDFS.
The key of the Mapper is NullWritable (null) and the value BytesWritable type. All the processing algorithms will be carried out in the Mapper.

D. *Reducer*

The Reducer portion is written in java, is a set of intermediate values which share a key to a smaller set of values.
The reducer contains the two phases:

1. Sorting: the reducer sorting the pixels of sub images as a key-value pair.
2. Shuffling: the reducer copies the sorted images from mapper using HTTP across the network.

The shuffling and sorting phases works simultaneously.

VII. ALGORITHMS USED IN REMOTE SENSING IMAGES

A. *Sobel Edge Detection Method*

The Sobel edge detector is used to detect edges and is based on applying horizontal and vertical filters in sequence. Both filters are applied to the image and summed to form the final result. Edges in images are areas with strong intensity contrasts. Detecting the edges in an image significantly reduces the amount of data and filters out useless information while preserving the important structural properties in an image.
The Sobel operator applies a 2-D spatial gradient measurement to an image and is represented by an equation. It is used to find the approximate gradient magnitude at each point in an input grayscale image. The discrete form of the Sobel operator can be represented by a pair of 3 × 3 convolution kernels, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). The Gx kernel highlights the edges in the horizontal direction, while the Gy kernel highlights the edges in the vertical direction. The overall magnitude of the outputs, |G|, detects edges in both directions and is the brightness value of the output image.

B. *Laplacian Edge Detection*

The Laplacian Edge Detection method is a second-order derivative of an image and it is applied by convolving the non-directional Laplacian filter. The second order derivative an edge will have a zero crossing in the region where there is the highest change in intensity. Therefore the location of the edge can be obtained by detecting the zero-crossings of the second-order derivative of the image and this is known as Laplacian filter which is an effective detector for non-sharp edges where the pixel intensity level change over space slowly. A single filtering kernel of different sizes (e.g. 3x3, 5x5, 7x7, etc.) that has low values (usually negative) in the middle of the kernel surrounded by positive values can be used as Laplacian edge detection.Because the Laplacian is an approximation of the second-order derivative of an image preserving the high frequency components, it is very sensitive to noise and therefore it is usually applied to an image that has first been smoothed using the Gaussian filter in order to suppress noises in the image.

C. *Canny Edge Detection*

The Canny edge detection is considered as the optimal and standard edge detector. This multi-step method which was developed by the canny,which aims to develop an optimal algorithms that satisfies three main criteria.The first one is 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.The second criterion is that detected edges should be as close as possible to the real edges. The third criterion is to have minimal number of response and edges should not be detected more than once. To satisfy these criteria, the algorithm can be performed in the following four separate steps.

1. *Smoothing* : This step involves smoothing the image using a Gaussian filter to suppress the noise and the degree of smoothing is controlled by the standard deviation σ of the Gaussian filter
2. *Gradient Magnitude and Direction*:The gradient magnitude of the image is computed using any of the gradient operators (e.g. Sobel, Roberts, Prewitt) and the direction of the gradient, the angle is rounded to the closest 0, 45, 90 or 135 degree angle.
3. *Non-maximum suppression*: From the image gradient the local maxima are identified as edges based on its direction and the non-maximum image intensities are suppressed.
4. *Thresholding by hysteresis*: Assuming that true edges are continuous, thresholding is done with hysteresis which requires upper and lower threshold.The upper threshold selects those edges that are strong. These edges are used to trace the weak edges while applying the lower threshold to suppress those edges that weak and not connected to the strong edges.After completion of this process, the final image output becomes a binary format.

VIII. SIMULATION USING VIRTUAL MACHINE

For simulating the software, we address 5 nodes Hadoop cluster configuration in details.5-nodes cluster’s configuration are similar but nodes are different. Five virtual machines are built on a server side. The 5-nodes cluster configuration information is shown in the following tables:

Table 1. Cluster Hardware Details

Name	Amount	Detailed
Name Node	1	1CPU*2.0GHz 4GB RAM
Data Node	4	1 CPU*2.0GHz 2GB
Hard Drive		30GB

Table 2. Cluster Software Details

Software	Version
Ubuntu	12.04
JDK	Jdk1.8.0_25
Hadoop	2.6.0

Table 3. Cluster Network Details

Node Name	Detailed
Master	192.168.0.1
Slave1	192.168.0.2
Slave2	192.168.0.3
Slave3	192.168.0.4
Slave4	192.168.0.5

In this paper, VM Box software is used to build cluster of virtual machines. VM Box virtual machine software is the leader company of data center virtualization solutions. The reason of using virtual machine is,it is relatively efficient to set up virtual Hadoop cluster to different scenarios; it software part can expand quickly and its very convenient to add or drop nodes, the hardware configuration can also be expand.For example,if a virtual machine is low on memory, the virtual machine can be turned off automatically again and again, then by using software settings we can increase virtual machine's memory. The total memory of all virtual machine nodes cannot exceed the actual memory size of the physical machine.

IX. EXPERIMENTAL RESULT AND ANALYSIS

In this paper, we analysis the result of some parallel image edge detection algorithms applied to remote sensing images available in the USGS website.

We mainly consider the time consumption with increasing the large volume of remote sensing data, and we try to show the difference in run time between a single PC implementation and the parallel Hadoop implementation.The different configurations of the cluster computer and single computer computing platforms used to obtain the experimental results.We implemented the image Sobel edge detection, Laplacian edge detection and Canny Edge Detection by using Java.The programs ran sucessfully when the size of the input image was large. These useful edge detection algorithms were chosen in the Hadoop project for their frequent use in remote sensing and for their diverse computational loads. We paid more attention on the processing time instead of the number of images processed.We compared the times taken by the single system and the multinode cluster to observe the speedup factors and finally analyzed the results.

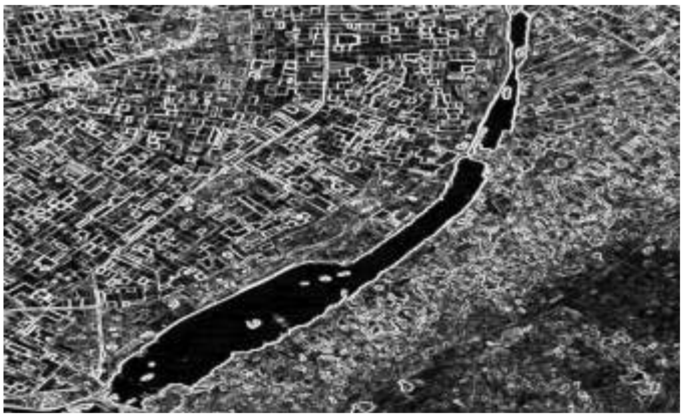
Fig 2: Sobel Edge Detection Sample Output

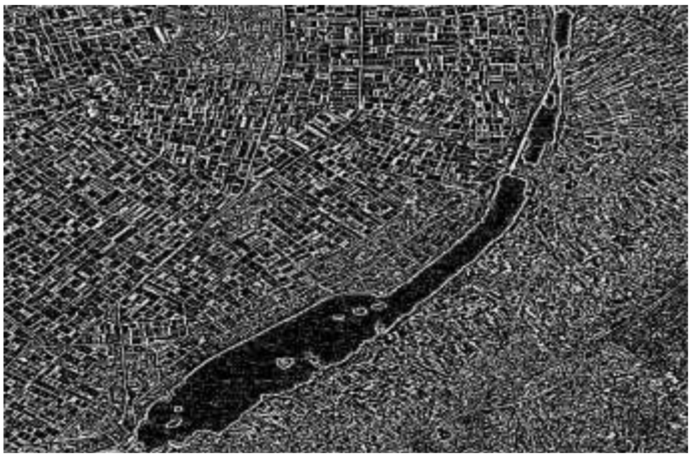
Fig 3: Laplacian Edge Detection Sample Output

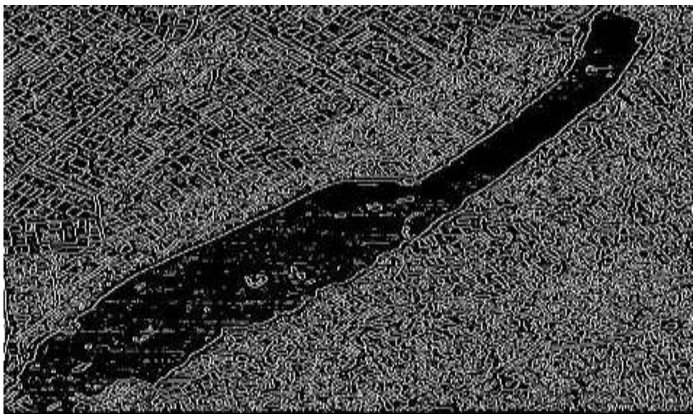
Fig 4: Canny Edge Detection Sample Output

X. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a case study for implementing parallel processing of remote sensing images in TIF format by using the Hadoop MapReduce framework. The experimental results and analysis have shown that the image processing algorithms can be effectively parallelized with acceptable run times when applied to large remote sensing images. A large number of satellite images cannot be processed efficiently in the sequential way. Hadoop installed on a cluster of computer in a parellel way to proved suitable to process TIF format images in large quantities. We have observed that this parallel Hadoop implementation is better suited for large data sizes ,when a computationally remote sensing applications is required.In the future, we might focus on using different image sources with different algorithms that can have a computationally intensive nature.

REFERENCES

[1] B. Li, H. Zhao, Z. H. Lv, "Parallel ISODATA clustering of remote sensing images based on MapReduce," in 2010 Int. Conf. Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC).

[2] T. White, Hadoop: The Definitive Guide. O'Reilly Media, Yahoo! Press, June 5, 2009.

[3] Y. Li and T. R. Bretschneider. Semantic-sensitive satellite image retrieval.IEEE Transactions on Geoscience and Remote Sensing , 45(4):853–860, April 2007.

[4] P. M. Atkinson and A. R. L. Tatnall. Neral networks in remote sensing. International Journal of Remote Sensing , 18(4):699–709, April 1997 .

[5] T. White, MapReduce: The Definitive Guide. O'Reilly Media, Yahoo! Press, June 5, 2009M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[6] Towards Large Scale Land-cover Recognition of Satellite Images Noel C. F. Codella, Gang Hua, ApostolNatsev, John R. Smith.