International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 3

An Approach for Effectively Handling Small-Size Image Files in Hadoop

Arjumand Ali
Department of Computer Science and Engineering
Jamia Hamdard, New Delhi, India

ABSTRACT

Small size file handling problem is a big challenge in Hadoop framework. Many approaches have been proposed and evaluated to deal with the small size file handling problem in Hadoop. File merging strategy is one of the most popular approaches used in literature. To deal with the small size file handling problem this paper evaluates a merging strategy in the domain of "Content-based Image Retrieval Systems (CBIRS)". CBIRS form a perfect application domain for evaluating solutions for small size file handling problem in Hadoop by incorporating a huge number of image files (small files).The approach used in this paper is shown to be efficient than the solution provided by HIPI (Hadoop Image processing Interface).

Keywords: *Hadoop, Data node, Name node, HDFS client, Map reduce, small-file problem, Content-based Image Retrieval, Histogram Intersection*

1. INTRODUCTION

Hadoop is among the most favorable high-performance java based open source distributed computing platform that was designed to store & process big data. Hadoop gives the best performance for handling large sized files & consists of two components i.e HDFS & Map Reduce. HDFS is the primary component of the Hadoop with a default data block size of 128 MB meant for managing & storing large sized files. When the "size of file" is much smaller than the default HDFS block size, the efficiency is degraded. HDFS supports master-slave architecture & follows writing once but reading many times pattern. One of the most important advantages of HDFS is data replication. Map Reduce is regarded as the heart of the Hadoop. Map reduce is a software framework, a programming model & a processing part of the Hadoop that makes use of computing resources & is used to process & generate large datasets in a reliable & fault tolerant manner. In map reduce, Hadoop program performs two separate & distinct tasks in which a sequence file is used for the purpose of input/output formats. Both job trackers & task trackers are incorporated in map reduce. It is not necessary to write map reduce jobs in java but Hadoop is implemented in Java.

Generally, in Hadoop deployment, we have three major types of machine roles. They are Name node, Data node & client machine.

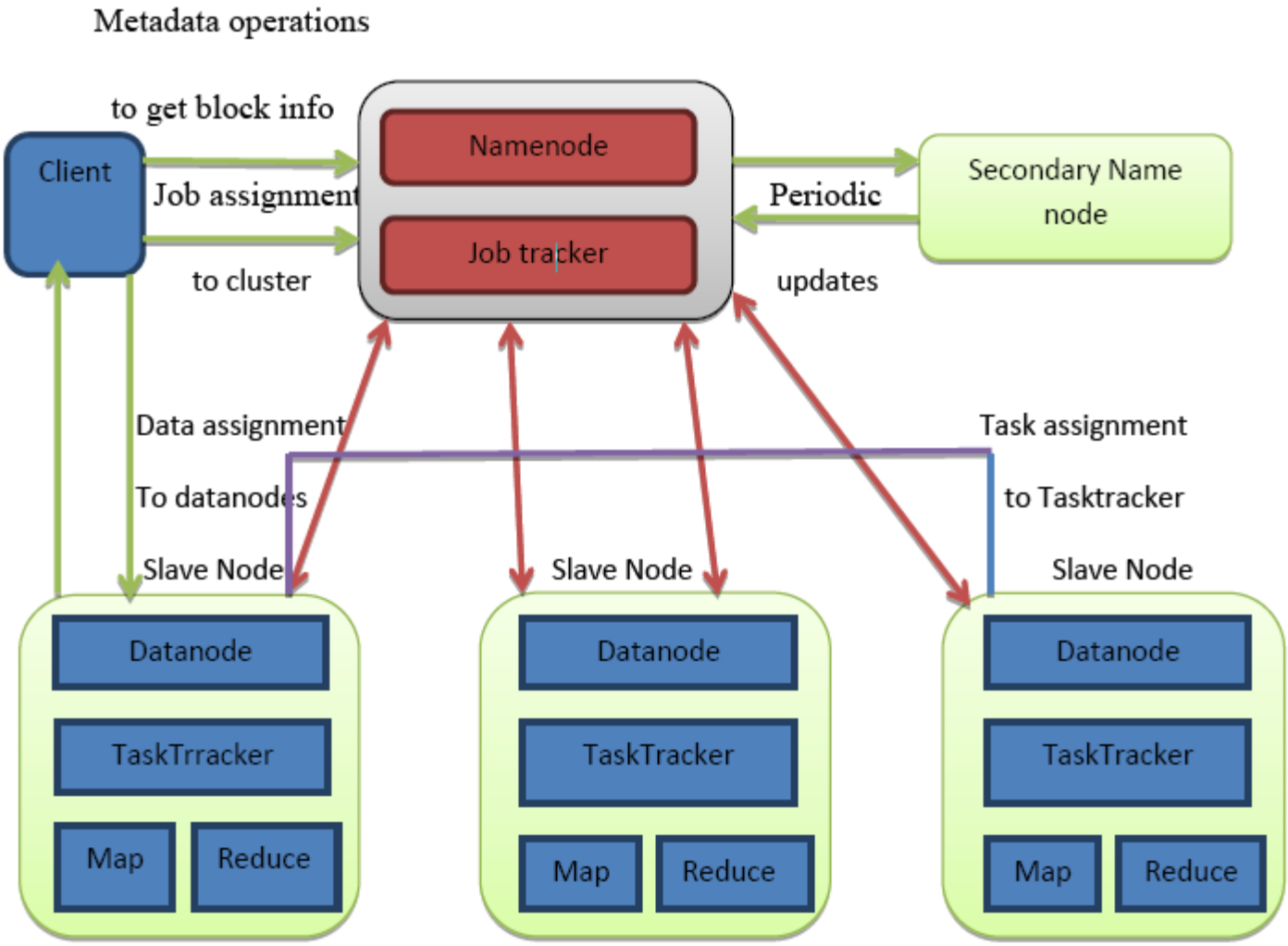

Figure 1 Hadoop Distributed File System architecture

1.1 Image Files:-Image files are standardized means of storing & organizing digital images that can be rasterized for use on a printer or computer display. There are hundreds of image file types & each file is created for different purposes & has its own pros & cons. The PNG, GIF & JPEG formats are most often used to display images on the internet. Image files also come under the category of small files in which block size is less than HDFS default block size & therefore we use H**adoop** in order to tackle such files.

1.2 Small File Problem:-We have many files which are much smaller than the HDFS default block size that is why we use H**adoop** in order to tackle such problems. Heavy consumption of Namenode memory, Degradation of Map-reduce performance, Queuing & huge no of JVM's are the main problems in small files. Small size file handling problem on Hadoop occurs mainly due to two issues i. e performance on map reduce reduces & high Name node memory utilization.

1.3 CBIR: For browsing, retrieving & searching digital images from a large database a computer system known as "Image retrieval System" is used. Some of the applications of image retrieval are remote sensing, crime prevention, fashion, architecture & medicine etc. "Text-based image retrieval (TBIR)" & "content-based image retrieval (CBIR)" systems are two types of Image retrieval systems. TBIR is having demerits of efficiency, time-consuming, loss of information & more expensive task. In order to overcome these problems, CBIR or QBIC systems are used. CBIR involves two steps: Feature extraction & matching.

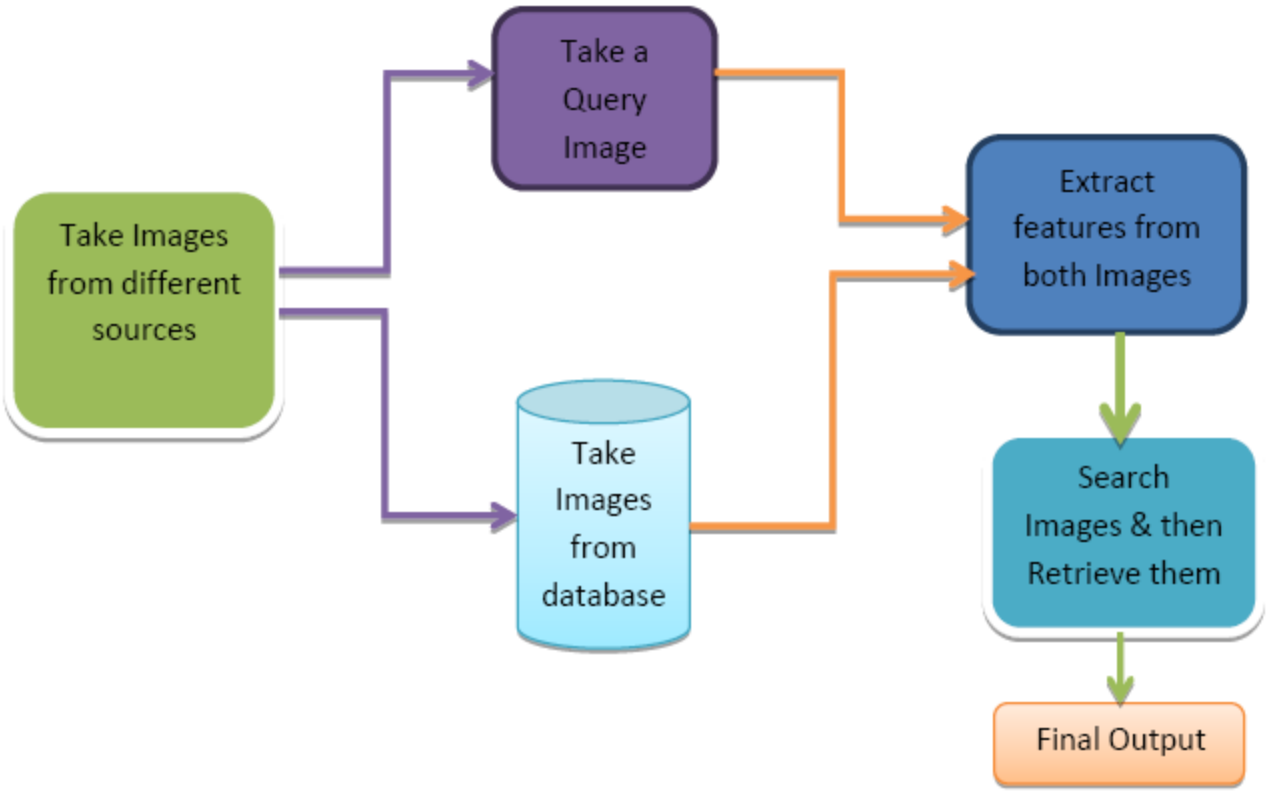

Figure 2 Structure of CBIR

2. PROBLEM IDENTIFICATION

Small size file handling problem is a major challenge in Hadoop framework. This paper explains two major objectives for handling small files.

Explore a merging strategy for small size file handling problem.

Use merging strategy for small files to show Performance evaluation for Content-based Image Retrieval Systems.

There are different techniques and solutions that have been proposed to deal with small size file handling problem in Hadoop. This paper shows the application of Hadoop in image processing by evaluating CBIRS to explore a solution to the small size file handling problem in Hadoop framework.

This paper has been divided into 6 sections. In Section 3 we discuss the related works. Section 4 presents the proposed approach, Section 5 shows the experimental results & Section 6 provides the conclusion of the work.

3. RELATED WORKS

Chethan. R et al. [1] proposed a separate algorithm for map & reducer in map reduce, gave the concept of merging strategy & also described a mathematical model. **Sushmitha et al.[2]** proposed an approach similar to 'merging' solution & avoids all those files to merge whose size is greater than the threshold but it was very time-consuming. So this paper made use of the map-reduce model in order to reduce time consumption, providing a minimum response time for batch analysis, handles sequence files & text files efficiently & reduces the time of executing & merging of files. **Kashmira P.Jayakar & Y.B.Gurav[3]** has proposed a solution called as EHDFS which consists of four operations file mapping, file merging, file extraction & prefetching. In this approach, for a combined file a file & block metadata is maintained by name node, and for accessing individual files, an indexing mechanism is proposed. Furthermore "Index prefetching" is incorporated to improve the I/O performance. **Gupta Bharti et al. [4]** this paper consists of the following five phases: merging strategy, local file strategy, fragmentation, caching & uploading of files to HDFS. The first phase is file merging strategy which is similar to the solution proposed by all the above authors. In the second phase, an index file is created for each original file which contains 4 parameters. In the third phase, merged files were partitioned in such a way so that no internal fragmentation occurs. In the fourth phase, Name node stores the information of the index file & merged file to avoid the overhead. The last phase is used for correlated & index files. **Dong, Bo, et al. [5]**

proposed an approach which mainly consists of two aspects: prefetching & file merging & the approach which is adopted by Blue Sky mainly consists of three layers: Business Process layer, User Interface layer & the Storage layer. **Chandrasekar S et al. [6]** proposed a similar approach as already proposed by Doug et al. by extending an HDFS known as EHDFS which provides an improved prefetching, indexing mechanism & consists of four techniques prefetching, file mapping, file merging & file extraction in order to improve an efficiency**. Mansoor Ahmad Mir & Jawed Ahmad [7]** proposed a solution similar to 'merging' in which an index table was maintained for each block. This paper also made use of Fetching in which the access time of data from memory is high as compared to from cache & for fast retrieval of information a caching mechanism was used. **Mr.Shubham Bhandari et al.[8]** this paper explained the architecture of NHAR to reduce the stress on Namenode & discussed the Configuring Hashtable of Name node for the NHAR in which each Name node is confined to three Hashtables. **Chatuporn Vorapongkitipun & Natawut Nupairoj [9]** proposed an approach based on HAR known as NHAR. NHAR enhances HAR to add additional files. This paper provides the ability to access smaller files in HDFS & also increases the memory usage of metadata. **Guru Prasad M SI et al.[10]** proposed two techniques i.e Map Combine Reduce & File Manager for handling small files in Hadoop. File Manager solves the memory stress on Namenode, manages the metadata, provides mutable property to HDFS files, distributes files to computing nodes & performs four functions in which a separate algorithm was proposed for each function. **J. Shashank et al. [11]** In case of querying an image from a database this paper describes the Content level access which is kept protected from every person as well as a database admin. CBIR deals with this type of issue. This paper also describes users issue about data privacy & provides experimental study, scalability & efficiency of the algorithm which is customizable for the structure of hierarchical data. **S. M. Patil et al. [12]** allows storing images in the database by its features & deals with technical, large & distributed collection of scientific images which contain complex information & can be retrieved using sophisticated, query-based semantics & precise measures of similarity. This paper also describes the information related to "low-level properties" of the image and produced the outputs. **S. Murali et al. [13]** proposed a method which gives the relationship between two pixels i.e a referenced pixel with its neighbor's pixel with respect to the direction & proposed an algorithm for retrieving images for CBIRS using local term patterns (LTP) for indexing & also defines a standard combination between local term patterns & binary term patterns by calculating a difference in gray level between referenced pixel & its neighbors. **Dewen Zhuang et al. [14]** proposed a relevance feedback method to improve performance, retrieval efficiency, reduces semantic gap & the storage of image signatures. **Swapnil et al. [15]** proposed a solution for retrieving "similar images" with respect to the query image from a database and from a large database of images in a secure, effective & efficient manner with the help of a technique known as Local Tera Patterns (LTP) by calculating a difference in gray level between the center pixels & its neighbors. **Hinge Smita et al. [16]** This paper made use of the map-reduce framework to retrieve & extract features from images mainly used for parallel processing & then returns the result in less time. **B.R. Kavitha et al. [17]** proposed an approach which consists of three stages of obtaining the final result of CBIRS. The First stage involves feature extraction & storage of images. The Second stage involves the implementation of algorithm & extraction of features from a query image. The last stage involves the retrieval of images in the order in which it matches with the properties of a query image. **K.Kranthi Kumar & Dr.T.Venu Gopal [18]** proposed matching & comparison algorithms to match & compare the features of one image with another image. To calculate the similarity between features algorithms were used by CBIR systems & a separate algorithm was used for each feature one for extraction & another for matching. Michael J .Swain & Dana H. Ballard **[19]** proposed Histogram Intersection in their article "color indexing". Ballard & Brown **[20]** proposed that there are "three opponent color axes" which are used for color axes of the histograms and are given below:

"rg" = "r – g"

"by" = "2 * b- r- g"

"wb" = "r +g +b"

where "r", "g" and "b" represent "red", "green" and "blue" signals and "rg", "by" and "wb" are the three opponent color axes used by the human visual system.

This paper makes use of merging strategy for evaluating "Content-Based Image Retrieval Systems" by using a technique known as "Histogram Intersection".

4. Proposed Approach

Since the above-stated approaches require a lot of overhead with respect to pre-processing so there is a need for an approach where there is less processing overhead & less communication cost. In this paper to achieve the same, we evaluate a Content-based Image retrieval system by implementing a merging strategy. Our approach also overcomes the overhead incorporated in Hadoop Image Processing Interface (HIPI).

Our proposed approach is divided into two stages:-

4.1 Algorithm for Stage 1:

1) Take input an Image dataset where each image is a small file.

2) Instead of processing each image file separately we extract paths to these small files & store those files into a single file.

3) This path file is rendered on Hadoop platform.

4) Images are read through this path file & content based features are extracted from the images which make up histograms for different files.

5) These histograms are stored as objects using Sequencefileoutputformat.

4.2 Algorithm for stage 2:

1) Extract the features from the input query image in a similar manner which also makes up a histogram.

2) This input query histogram is compared with the stored histograms through histogram Intersection process.

3) Those histograms whose comparison results are specific to a given threshold are the output results.

5. EXPERIMENTS

No of Image files	No of mappers	No of bytes read & written	Map Input & Output Records	Input Split bytes & spilled Records	Total committed heap usage(bytes)
300	Total 4. 2 mappers for first level configuration & 2 for second level configuration	88250 360600	100 0	112 0	11867648

The dataset used to perform this experiment is BSDS300 which stands for "Berkeley Segmentation dataset" consisting of 300 images. The images are divided into a "test set of 100 images" and a "training set of 200 images". The half of the segmentation was obtained from presenting grayscale image and other half were obtained from presenting the color image. This experiment was done on Hadoop 2.0 with Cent Operating system 7, 8GB RAM, Intel i7 core, 1TB hard disk, Java JDK 1.8.0 and 2.00 GB processor. For performing image processing tasks in the distributed environment a library for Hadoop framework called HIPI is used which provides application programming Interfaces(API's) in which multiple files are read by a single mapper where after each culling stage during processing one mapper reads one image file so the problem is still there. Instead of processing each image file separately our approach extracts paths to these small files, stores those files into a single file and renders them on Hadoop platform. Images are read through this path file by a single mapper and content based features are extracted from the images which make up histograms for different image files and these histograms are stored as objects using "SequenceFileoutputformat" which increases the efficiency and also solves the problem of classical HIPI. Since multiple image files are read by a single mapper, the map-reduce paradigm can be used efficiently.

6. Conclusion and Future Scope

CBIRS is a challenge. Hadoop Platform is best suited for handling large sized files. Image files are small files and in order to analyze the contents of various

images, histograms are made, which store the color information of the images and outputs them as sequence file output format i.e. the histograms are stored as objects. The input query image is processed in the same manner making up a histogram. The input query histogram is compared with the stored histogram through histogram Intersection process. In Small size file handling problem & HIPI one mapper reads one image file which is a limitation. So this paper makes use of Sequencefileoutputformat in which multiple files are read by a single mapper due to which the time for reading from the hard disks will be less and the processing time will get reduced to a greater extent.

Audio and video files also come under the category of small files, as a future work, we can explore these types of files too since these files will also suffer performance issues which are faced by small files stored in HDFS.

REFERENCES

1. Scholar, U. G.A Selective Approach for Storing Small Files in Respective Blocks of Hadoop.
2. Roshini, R., & Raikar, M. Map Reduce based Analysis of Live Website Traffic Integrated with Improved Performance for Small Files using Hadoop.
3. Jayakar, K. P., & Gaurav, Y. B. Efficient Way for Handling Small Files using Extended HDFS.International Journal of Computer Science & Mobile Computing in June 2014.
4. Gupta, B., Nath, R., & Gopal, G.An Efficient Approach for Storing and Accessing Small Files with Big Data Technology. *International Journal of Computer Applications*, in the year 2016.
5. Dong, Bo, et al."A Noval approach to improving the efficiency of storing & accessing of small files on Hadoop: a case study by ppt files." International Conference on Services Computing, in the year 2010.
6. Chandrasekar, S., et al. "A novel indexing scheme for efficient handling of small files in Hadoop distributed file system." International Conference on Computer Communication & Informatics (ICCI), in the year 2013.
7. Mir, M.A., & Ahmad, J. "An Optimal Solution small file problem in Hadoop. International Journal of Advanced Research in Computer Science," in the year 2017.
8. Mr. Shubham Bhandari et al.An approach to solving a small file problem in Hadoop by using Dynamic Merging & Indexing Scheme. International Journal on Recent & Innovation Trends in Computing & Communication.
9. ChatupornVorapongkitipun & Natawut Nupairoj "Improving performance of small-file accessing in Hadoop" 11th International Joint Conference on Computer Science & Software Engineering (JCSSE), in the year 2014.
10. Prasad, G., Nagesh, H.R., & Deepthi, M. Improving the Performance of Processing for Small files in Hadoop: A Case Study of Weather Data Analytics.International Journal of Computer Science & Information Technology, in the year 2014.
11. J. Shashank, "Content-Based Image Retrieval Using Color, Texture & Shape features," International Conference on Advanced Computing & Communications(ICACC), in the year 2007.
12. Shankar M. Patil "Content-Based Image Retrieval using Color, Texture & Shape," International Journal of Computer Science & Engineering Technology (IJCSET), in the year 2012.
13. Murala S., Maheswari R.P., & Balasubramanian, R., "Local Tetra Patterns: A New Feature Descriptor for CBIR" IEEE Transactions on Image Processing, in the year 2012.
14. Kusuma,B & Megha.P.Arakeri."Survey on Content-based Image Retrieval using MapReduce over Hadoop."International Journal of Advances in Electronics & Computer Science, in the year 2015.
15. Smita, H., Monika, G., & Shraddha, C. Content-Based Image Retrieval Using Hadoop Map Reduce.International Journal of Computer Science trends & Technology, in the year 2014.
16. Kavitha, B.R., Kumaresen, P.,& Govindaraj, R. Content-Based Image Retrieval using Hadoop & HIPI, on 24th Dec 2017.
17. Kavitha, B.R., Kumaresan, P., & Govindaraj, R.Content-based Image Retrieval using Hadoop and HIPI.
18. K.Kranthi Kumar & Dr. T. Venu Gopal. "CBIR: Content-Based Image Retrieval."A Conference paper in the year 2014.
19. Swain, M. J., & Ballard, D.H. Color indexing. International Journal of computer vision,in 1991.
20. Ballard D H., & Brown, C .M. Computer vision. Prentice Hall Professional Technical reference in 1982.