ISSN 2455-9954
International Journal Of Emerging Technology and Computer Science

Distributed Image Processing using Different Techniques in Hadoop

Mrs. Parvati Bhadre, Mrs. Vishnupriya G. S., Rajasree R. S.
paru.nadgeri@gmail.com, vishnupriyag@gmail.com, rajasreecse@gmail.com,
Assistant Professor D Y Patil College of Engineering, Akurdi.

Abstract

With the rapid growth of social media, the quantity of pictures being uploaded to the net is exploding. Huge quantities of pictures are shared through multi-platform services such as Snapchat, Instagram, Facebook and WhatsApp; recent studies estimate that over 1.8 billion photos are uploaded each day. However, for the most part, applications that make use of this large information have yet to emerge. Most current image process applications, designed for small-scale, native computation, don't scale well to web-sized issues with their massive necessities for machine resources and storage. The emergence of process frameworks like the Hadoop and MapReduce platform addresses the matter of providing a system for computationally intensive processing and distributed storage. However, to find out the technical complexities of developing helpful applications using Hadoop needs an outsized investment of your time and skill on the a part of the developer. As such, the pool of researchers and programmers with the numerous skills to develop applications that can use massive sets of pictures has been restricted. To handle this we've developed the Hadoop Image process Framework that provides a Hadoop-based library to support large-scale image process. The most aim of the framework is to permit developers of image process applications to leverage the Hadoop MapReduce framework while not having to master its technical details and introduce a further source of quality and error into their programs.

Introduction

With the spread of social media in recent years, an oversized quantity of image information has been accumulating. Once process this huge information resource has been limited to single computers, computational power and storage ability quickly become bottlenecks. Alternately, processing tasks can usually be performed on a distributed system by dividing the task into many subtasks. The ability to parallelize tasks permits for scalable, efficient execution of resource-intensive applications. The Hadoop Map Reduce framework provides a platform for such tasks. When considering operations like face detection, image classification and different types of process on pictures, there are a unit limits on what are often done to boost performance of single computers to form them able to method data at the dimensions of social media. Therefore, the benefits of parallel distributed process of an oversized image dataset by exploitation the procedure resources of a cloud computing surroundings ought to be thought of. In addition, if procedure resources are often secured simply and comparatively inexpensively, then cloud computing is appropriate for handling massive image information sets at very low price and increased performance. Hadoop, as a system for process massive numbers of pictures by parallel and distributed computing, looks promising. In fact, Hadoop is in use everywhere the world. Studies exploitation Hadoop are performed, addressing text information files, analysing large volumes of DNA sequence data[10], changing the info of a large variety of still pictures to PDF format, and completing feature selection/extraction in astronomy. These examples demonstrate the quality of the Hadoop system, which may run multiple processes in parallel for load levelling and task management.

Most of the image process applications that use the Hadoop MapReduce framework are extremely complicated and impose a staggering learning curve. The overhead, in computer programmer time and expertise, needed to implement such applications is cumbersome. To address this, we tend to gift the Hadoop Image process Framework, that hides the extremely technical details of the Hadoop system and permits programmers UN agency will implement image processing algorithms however who have no specific experience in distributed systems to even so leverage the advanced resources of a distributed, cloud-oriented Hadoop system. Our framework provides users with quick access to large-scale image information, smoothly enabling rapid prototyping and flexible application of complicated image process algorithms to terribly massive, distributed image databases.

Literature Survey

With the rapid usage increase of on-line picture storage and social media on sites like Facebook, Flickr and Picasa, a lot of image knowledge is obtainable than ever before, and is growing every day. Each minute twenty seven,800 photos are uploaded to Instagram,[6] whereas Facebook receives 208,300 photos over constant time-frame. This alone provides a supply of image data which will scale into the billions. The explosion of obtainable pictures on social media has motivated image process analysis and application development which will benefit of very large image information stores.

White et.al [5] presents a case study of classifying and clustering billions of normal images mistreatment MapReduce. It describes a picture pre-processing technique to be used in a sliding-window approach for visual perception. Pereira et.al [3] outlines a number of the restrictions of the MapReduce model when managing high-speed video coding, namely its dependence on the NameNode as one purpose of failure, and also the difficulties inherent in generalizing the framework to suit explicit problems. It proposes associate alternate optimized implementation for providing cloud-based IaaS (Infrastructure as a Service) solutions.

Lvet.al [9] describes mistreatment the k-means algorithmic rule in conjunction with MapReduce and satellite/aerial photographs so as to find totally different components supported their color. Zhang et.al [7] presents strategies used for process sequences of magnifier pictures of live cells. The photographs are comparatively little (512x512, 16-bit pixels) keep in ninety MB folders, the authors encountered difficulties regarding fitting into Hadoop DFS blocks with were resolved by custom Input Format, Input Split and Record Reader categories. Powell et.al [5] describes however National Aeronautics and Space Administration handles image process of celestial pictures captured by the Mars equipment and rovers. Clear and apothegmatic descriptions are provided concerning the segmentation of gigapixel pictures into tiles, however the tiles are processed and the way the image processing framework handles scaling and works with the distributed process. Wang, Yinhai and McCleary[6] discuss speeding up the analysis of tissue microarray pictures by substituting human knowledgeable analysis for automatic process algorithms. Whereas the photographs were gigapixel-sized, the content was simply divided and there was no need to analyse all of a picture right away. The work was all done on a specially-built high performance computing platform mistreatment the Hadoop framework.

Bajcsy et.al [1] gift a characterization of 4 basic terabyte-size image computations on a Hadoop cluster in terms of their relative efficiency according to a modified Amdahl's Law. The work was motivated by the actual fact that there's a scarcity of normal benchmarks and stress tests for large-scale image process operations on the Hadoop framework. Moise et.al [9] outlines the querying of thousands of pictures in one run mistreatment the Hadoop MapReduce framework and also the eCP algorithmic rule. The experiment performs a picture search on a hundred and ten million pictures collected from the online using the Grid 5000 platform. The results are evaluated so as to grasp the simplest practices for standardization Hadoop MapReduce performance for image search.

Paper	**Year/authors**	**Technique**	**Conclusion**
Web-scale computer vision using mapreduce for multimedia data mining.	2010 Brandyn White, Tom Yeh, Jimmy Lin, and Larry Davis	classifying and clustering billions of normal images mistreatment MapReduce	a picture pre-processing technique to be used in a sliding-window approach for visual perception
R. Pereira, M. Azambuja, K. Breitman, and M. Endler.	2010 An architecture for distributed high performance video processing in the cloud.	It proposes associate alternate optimized implementation for providing cloud-based IaaS (Infrastructure as a Service) solutions.	outlines a number of the restrictions of the MapReduce model when managing high-speed video coding
Terabytesized image computations on hadoop cluster platforms	2013 P. Bajcsy, A Vandecreme, J. Amelot, P. Nguyen, J. Chalfoun, and M. Brady.	characterization of 4 basic terabyte-size image computations on a Hadoop cluster in terms of their relative efficiency	there's a scarcity of normal benchmarks and stress tests for large-scale image process operations on the Hadoop framework.
Case study of scientific data processing on a cloud using hadoop.	2010 Chen Zhang, Hans De Sterck, Ashraf Aboulnaga, Haig Djambazian, and	Presents strategies used for process sequences of magnifier pictures of live cells. The	encountered difficulties regarding fitting into Hadoop DFS blocks

	Rob Sladek.	photographs are comparatively little (512x512, 16-bit pixels) keep in ninety MB folders	with were resolved by custom Input Format, Input Split and Record Reader categories

IMAGE PROCESSING CLOUD

A. HDFS

Hadoop platform provides distributed filing system (HDFS) that supports great deal of information storage and access. Hadoop MapReduce programming model supports multiprocessing data supported the widely-used map-and-reduce parallel execution pattern. so as to support the multiple language requirements in image process domain decide Hadoop streaming programming model by redaction customary input and output, and stream knowledge to applications written with completely different programming languages. Moreover, the streaming model is also simple to correct during a standalone model that is essential to test a rule before getting to large-scale. The image process application execution surroundings with MapReduce on Hadoop is shown in Figure a pair of. On the left side, an oversized range of pictures area unit hold on in HDFS, which area unit distributed across the cluster with 128MB in concert block. These pictures area unit split by Hadoop MapReduce engine with made-to-order InputFormat, and area unit distributed to giant number of mappers that execute image process applications to the allotted pictures. The results could also be incorporated by the reducer that exports the results to personalized OutputFormat class to finally save the outputs. Since great deal information are transferred among split, mappers and reducers, it is very important to stay knowledge neighbourhood to attenuate network traffic. All mappers are launched on the node wherever the processed pictures are physically hold on[1].

B. Mapper and Reducer

Most of work for programming in Hadoop is to divide algorithms into mapper and Reducer, and embed and implement them in them severally. In Hadoop streaming mode, the main difference with different modes is that the I/O process in Mapper and Reducer. Each mapper and Reducer may solely get Key/Value from Stdin and output results through Stdout. A common I/O category named CommonFileIO was designed to handle totally different kind information sources, as well as normal native files, Stdin/Stdout and HDFS file on Hadoop. The frequently used file system interfaces were provided, like open, read/write and close and a lot of. Mapper and Reducer works as freelance image process applications with input and output handled by Stdin and Stdout. By using Hadoop streaming model, sizable amount of Mappers or Reducers execute in parallel.

MapReduce is recognized as a popular framework to handle huge data amount in the cloud environment due to its excellent scalability and fault tolerance. Application programs based on MapReduce can work on a huge cluster of thousands of desktops and reliably process Peta-Bytes data in parallel. Owing to this, time efficiency successfully gains a desirable improvement. Until now, MapReduce has been widely applied into numerous applications including data anonymization, text tokenization, indexing and searching, data mining, machine learning, etc.

Aimed at achieving higher time efficiency on the associated applications, recent endeavors involve many industry giants to make efforts by leveraging MapReduce. For example, Yahoo has been working on a couple of real-time analytic projects, including S4 and MapReduce Online. In addition, IBM has been devoted to developing real-time products such as InfoSphere Streams and Jonass

Entity Analytics software used to analyze stream data more accurately. Despite that these frameworks have successfully implemented the efficient processing of text data and stream data; they do little contribution to image processing field. Motivated by these cases achievements and restrictions, aim of this technique to provide an effective processing framework for big image data by the utilization of the cloud computing ability that MapReduce provides[1].

There is a novel effective distributed framework named Image Cloud Processing (ICP) which is dedicated to offering a reliable and efficient model for vision tasks. The core design of ICP is to utilize the affluent computing resources provided by the distributed system so as to implement effective parallel processing. The elegant distributed processing mechanism that ICP contains is defined from two comprehensive perspectives:

a) Efficiently processing those static big image data already stored in the distributed system, such as the task of image classification, image retrieval, etc. that do not demand immediate response to the users but an efficient processing instead;

b) Timely processing that dynamic input which needs to be processed immediately and return an immediate response to the users, especially for the requests from the user terminal, e.g., the image processing software in the users' laptop / desktop.

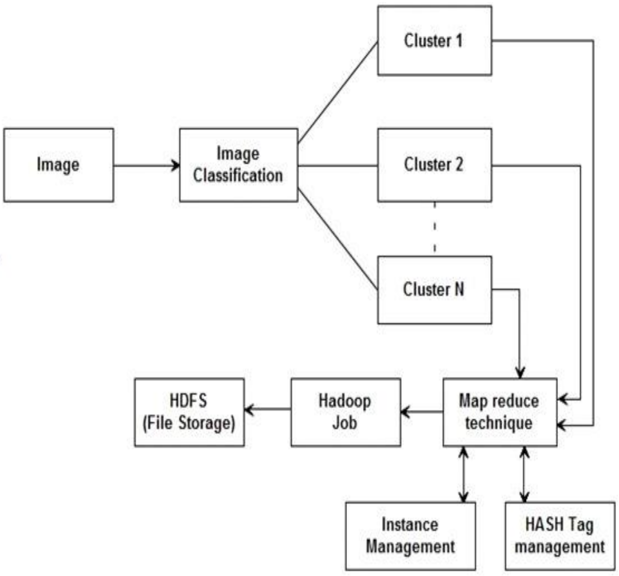

Figure 1 Architecture

C. Random sample consensus (RANSAC)

RANSAC is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are to be accorded no influence on the values of the estimates. Therefore, it also can be interpreted as an outlier detection method. It is a nondeterministic algorithm in the sense that it produces a reasonable result only with a certain probability, with this probability increasing as more iteration are allowed. A basic assumption is that the data consists of "inliers" i.e., data whose distribution can be explained by some set of model parameters, though may be subject to noise, and "outliers" which are data that do not fit the model. The outliers can come, e.g., from extreme values of the noise or from erroneous measurements or incorrect hypotheses about the interpretation of data. RANSAC also assumes that, given a (usually small) set of inliers, there exists a procedure which can estimate the parameters of a model that optimally explains or fits this data [12].

The RANSAC algorithm is a learning technique to estimate parameters of a model by random sampling of observed data. Given a dataset whose data elements contain both inliers and outliers, RANSAC uses the voting scheme to find the optimal fitting result. Data elements in the dataset are used to vote for one or multiple models. The implementation of this voting scheme is based on two assumptions: that the noisy features will not vote consistently for any single model (few outliers) and there are enough features to agree on a good model (few missing data). The RANSAC algorithm is essentially composed of two steps that are iteratively repeated:

a) In the first step, a sample subset containing minimal data items is randomly selected from the input dataset. A fitting model and the corresponding model parameters are computed using only the elements of this sample subset. The cardinality of the sample subset is the smallest sufficient to determine the model parameters.

b) In the second step, the algorithm checks which elements of the entire dataset are consistent with the model instantiated by the estimated model parameters obtained from the first step. A data element will be considered as an outlier if it does not fit the fitting model instantiated by the set of estimated model parameters within some error threshold that defines the maximum deviation attributable to the effect of noise [12].

The set of inliers obtained for the fitting model is called consensus set. The RANSAC algorithm will iteratively repeat the above two steps until the obtained consensus set in certain iteration has enough inliers. The input to the RANSAC algorithm is a set of observed data values, a way of fitting some kind of model to the observations, and some confidence parameters. RANSAC achieves its goal by repeating the following steps:

1) Select a random subset of the original data. Call this subset the hypothetical inliers.

2) A model is fitted to the set of hypothetical inliers.

3) All other data are then tested against the fitted model. Those points that fit the estimated model well, according to some model-specific loss function are considered as part of the consensus set.

4) The estimated model is reasonably good if sufficiently many points have been classified as part of the consensus set.

5) Afterwards, the model may be improved by re estimating it using all members of the consensus set.

Summary

Main goal is to explore the feasibility and performance of using Hadoop system to process large number of pictures, huge size of pictures or videos. Effective processing framework nominated Image Cloud Processing (ICP) to powerfully cope with the data explosion in image processing fieldHowever, there also are some problems have to be compelled to be thought-about and addressed in future work.

The first issue is the drawback of information distribution. As stated within the previous section, Hadoop is nice at handling big data. The acceleration isn't apparent whereas attempting to method many tiny pictures scattered across multiple nodes. Even the SequenceFile couldn't solve this drawback efficiently. Our next plan is making an attempt to store image files in HBase. HBase might handle random, realtime reading/writing access of huge knowledge.

References

[1] P. Bajcsy, A Vandecreme, J. Amelot, P. Nguyen, J. Chalfoun, and M. Brady. Terabytesized image computations on hadoop cluster platforms. In Big Data, 2013 IEEE International Conference on, pages 729–737, Oct 2013. [2] C.-I. C. Hsuan Ren and S.-S. Chiang, “Real-Time Processing Algorithms for Target Detection and Classification in Hyperspectral

Imagery," IEEE Transactions on Geoscience and Remote Sensing, vol. 39, no. 4, 2001, pp. 760–768.

[3] R. Pereira, M. Azambuja, K. Breitman, and M. Endler. An architecture for distributed high performance video processing in the cloud. In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, pages 482–489, July 2010

[4] L. L. Chris Sweeney and J. L. Sean Arietta, "HIPI: A hadoop image processing interface for image-based map reduce tasks," pp. 2–3, 2011.

[5] Brandyn White, Tom Yeh, Jimmy Lin, and Larry Davis. Web-scale computer vision
using mapreduce for multimedia data mining. In Proceedings of the Tenth International Workshop on Multimedia Data Mining, MDMKDD '10, pages 9:1–9:10, New York, NY, USA, 2010. ACM.

[7] Chen Zhang, Hans De Sterck, Ashraf Aboulnaga, Haig Djambazian, and Rob Sladek. Case study of scientific data processing on a cloud using hadoop. In DouglasJ.K. Mewhort, NatalieM. Cann, GaryW. Slater, and ThomasJ. Naughton, editors, High Performance Computing Systems and Applications, volume 5976 of Lecture Notes in Computer Science, pages 400–415. Springer Berlin Heidelberg, 2010.

[8] L. Dong, J. Su, and E. Izquierdo, "Scene-oriented hierarchical classification of blurry and noisy images," IEEE Trans. Circuits Syst.
Video Technol., vol. 21, no. 5, pp. 2534–2545, May 2012.

[9] L. Dong and E. Izquierdo, "A biologically inspired system for classification of natural images," IEEE Trans. Image Process., vol. 17, no. 5, pp. 590–603, May 2007.

[10] X. Tian, D. Tao, X. S. Hua, and X. Wu, "Active reranking for web image search," IEEE Trans. Image Process., vol. 19, no. 3, pp. 805–820, 2010.

[11] Y. Lin, et al., "Large-scale image classification: Fast feature extraction and SVM training," in Proc. IEEE Conf. Comput. Vis. Pattern
Recognit., 2011, pp. 1689–1696.

[11] "Intel Distribution of Hadoop," http://hadoop.intel.com/, [Retrieved: May, 2014].

[12] RANSAC algorithm: http://wikipedia/ransac.com/