Volume 4, Issue 12, December 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

HBase Cloud Research Architecture for Large Scale Image Processing

R. Saraswathy, P. Priyadharshini, P. Sandeepa
Assistant Prof. Dept. of Computer and and Engineering,
N.S.N. College of Engineering and Technology, India

Abstract - In this information world there is enormous amount of data flow between various Medias at very high rate. This data is categories as "Big Data". Most of this generated data are images and videos. The volume and the heterogeneity of data with the speed it is generated, makes it difficult for the present computing infrastructure to manage Big Data. Besides, image processing algorithm posses' great demand on storage and computation power. This paper aims to combine cloud computing and big data analysis technology to provide image processing cloud infrastructure and HBase based big data processing engine to satisfy the needs. Thus it is a great challenge not only to store and manage the large volume of data, but also provide solution to improve performance and scalability.

Keywords - Cloud Computing, Map Reduce, HBase, Image Processing, Apache Cloud Stack.

I. INTRODUCTION

We have entered the so-called big data generation where huge data are generated each and every day. Big data are developed by digital processing, mobile devices, Internet, social media computer systems and different types of sensors. Most of these new generated big data depends on images and videos. Big data analysis requires extensible computing power and practiced data mining statistics, machine learning, and pattern recognition capabilities. It is exaggerative in image processing domain since the video and image processing algorithms become more and more complicated, which demands even more power in computation. Some of these image processing requires even real-time processing capability. It is essential to create a domain specific cloud for image processing research in order think these challenging requirements.

Image processing research and education are used to support many fields. Image processing widely used in industries. The modern computer architectures have evolved to be more difficult, and continuously becomes a more challenge rather than help for general educators and researchers that use image processing technology. To utilize large scale computing resources image processing researchers will show scalability challenges, Hybrid parallel programming challenges of creating code for modern computer hardware configurations with multi-level parallelism, For example, cluster based on multicore processor nodes. It is not only difficult for researchers to implement their algorithms using existent programming environment but also very challenging to them to reuse and share the existing research results since these results are largely dependent on OS, libraries, and existing underlying architectures.

To fill the gap between more complicated modern architectures and highly emerging image processing algorithms for big data, our image processing with cloud project aims to produce a high-productivity and high-performance image processing research environment integrated within a cloud computing infrastructure. The cloud will not only provide sufficient storage and computation power to image processing researchers, but it also provides a open and shared environment to share the knowledge, education materials and research algorithms. By leveraging the cloud computing and big data processing technology, our design is to help the software and hardware complexity from researchers, so that they can focus on designing innovative image processing algorithms, instead of taking care of under-lining software and hardware details.

HBase is very different from traditional relational databases like SQL, MySQL, Post greSQL, Oracle, etc in how it's architected and the features that it provides to the applications using it. HBase trades off some of these features for scalability and a flexible schema. This also translates into HBase having a very different data model. Designing HBase tables is a different ballgame as compared to relational database systems.

II. RELATED WORK

The Images can be processed in parallel by using HBase platform. The major difference between our work and others is that our solution afford a PaaS and supports the various languages in implementing image processing algorithms. HIPI [3] is one of them that is similar to our work. In contrast to our work, HIPI [3] creates an interface for integrating multiple image files into a single large file in order to overcome the constraint of handling numerous small image files in HBase. The input type used in HIPI is referred to as a HipiImageBundle (HIB). A HIB is a set of images

combined into one large file along with some metadata describing the layout of the images. HIB is similar with HBase sequence file input format, but it is more customizable and mutable. However, users are required to modify the image storage using HIB, which creates additional overhead in programming. In our work, we make the image storage transparent to users, and there is no additional programming overhead for users to handle image storage.

HBase Mapreduce for Remote Sensing Image Analysis [6] aims to find an effective programming method for customized processing within the HBase MapReduce framework. It also uses the entire image as InputFormat for HBase, which is similar with our solution. However, the work only supports Java so that all mapper codes need to be written only in Java. By Comparing with the solution thus produced, the performance is not as good as the ours since we use native C++ implementation for OpenCV.

Parallel Image Database Processing with MapReduce and Performance Evaluation in Pseudo Distributed Mode [6] performs parallel distributed processing of a video database by using the computational resource in a cloud environment. It uses video database to store multiple sequential video frames, and uses Ruby as programming language for Mapper, thus runs on HBase with streaming mode . As a result, our platform is designed to be more flexible and supports multiple languages.

Large-scale Image Processing using MapReduce [2] try to investigate the feasibility of using MapReduce model for doing large scale image processing. It packaged large number of image files into several hundreds of Key-Value collections, and split one huge image into smaller pieces. It uses Java Native Interface(JNI) in Mapper to call OpenCV C++ algorithm. Same with the above work, this work only supports a single programming language with additional overhead from JNI to Mapper.

III. PROPOSED WORK

HBase Cloud Architecture

The HBase cloud computing infrastructure is built using several clusters together. Apache cloudstack is used to provide infrastructure as a service (iaas) to farm the virtual machines and to provide platform as a service (paas). HBase distributed cluster is used which store and process big data. the image processing cloud is built by integrating opencv library on the HBase cluster.

The infrastructure consists of three major components:

1. Cloud computing service portal built by Large machines number of Virtual Machines
2. Cluster of high performance is built to support high performance computing and big data processing.
3. Cloud data storage is used to support data access and storage.

The Cloud delivers suitable virtual machine from the VM farm to fulfill the powers needed for computing by submitting the works to the High performance Cluster and builds a bridge between complicated architecture and the end-users.

A. HBase Virtual machine farm cloud

A virtual machine farm based on Apache Cloud-Stack on top of an 56 nodes dual-core IBM cluster, and a new small Dell cluster with three 32 CPU cores servers, and one GPGPU server with 48 CPU cores and 1 NVIDIA Fermi GPGPU. To provide highly scalable IaaS cloud computing platform, Apache CloudStack is used. The goals of the HBase cloud are to provide IaaS and PaaS with customized services, to share resources, to facilitate teaching, and to allow faculty and students in different groups/institutions to share their research results and enable deeper collaborations. The CloudStack will create virtual machines and allocate resources for satisfying the user request.

B. Image Processing Cloud

The image processing cloud is built by combining the image processing library OpenCV with HBase platform to deliver PaaS specifically for image processing. The following describes the two major components.

- **HBase Cluster:** To provide PaaS the HBase big data processing framework on the bare-metal HPC cluster is installed within HBase Cloud. The HBase cluster consists of one 8-node HP cluster with 16-core and 128GB memory each and a 24-node IBM GPGPU cluster with 16-core and one Nvidia GPU in each node, and connected with Infinite Band interconnection. The Intel HBase Distribution based on Apache HBase software stack is installed, which is a framework that is designed to store and process big data on large-scale distributed systems with simplified parallel programming models. It consists of HBase common utilities, HBase Distributed File System (HDFS) for high-throughput and fault tolerance data access, HBase Yarn for job scheduling and resource management and HBase MapReduce based on a simple parallel pattern for parallel processing engine.

- **OpenCV Image Processing Library:** We selected the widely-used OpenCV (Computer Vision) [9] library as the base image processing library integrated with our image processing cloud. OpenCV is an open source library written in C++, and it also has Java and Python interfaces supporting Windows, Linux, Mac OS, iOS and Android. It is optimized and parallelized for multicores and accelerators using OpenCL. We installed the library on the HBase cluster to enable image processing capability with Map Reduce parallel programming model.

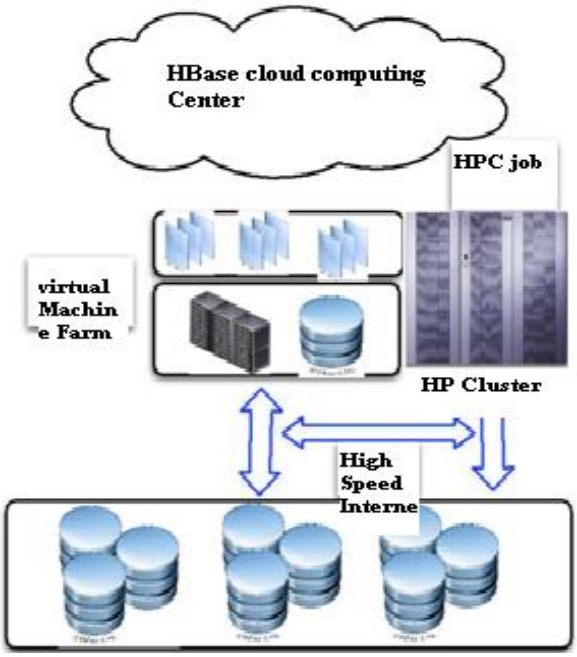

Figure1. HBase Cloud and HPC Cluster for Big Data Processing

Thus the scalable image processing cloud is implemented by combining these two components.

C. HBase Architecture

HBase is distributed database and is an open-source, NoSQL, distributed, column-oriented store that runs on top of HDFS and is developed as part of Apache's Hadoop project . HBase is really more a "Data Store" than "Data Base". Here the data is stored as a sections of columns and it consists of Table name, row key, column family, columns, time stamp. The row is uniquely identified with the help of row keys and time stamp and the column family are static and columns are dynamic. If the application with variable schema and each row is slightly different then HBase is preferable.

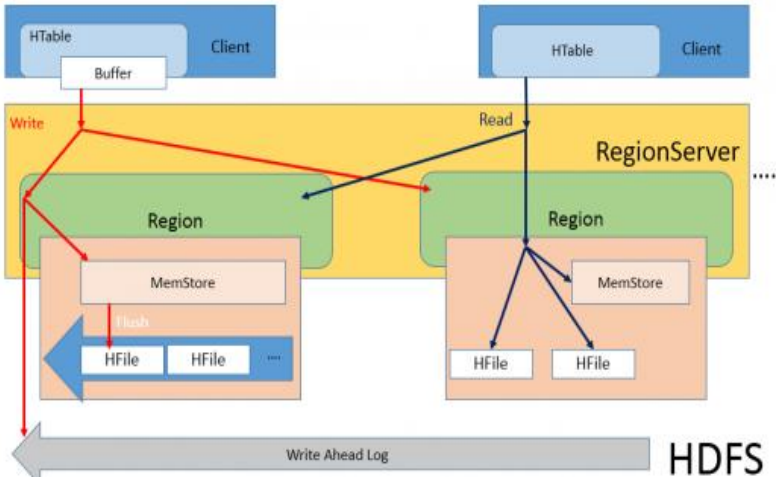

Figure2. HBase Architecture

In HBase, there are three important components: Master, Region server and Zoo keeper and the other components are Memstore ,WAL and HFile.As HBase runs on top of HDFS, it utilizes the Master-Slave architecture in which the HMaster will be the master node and the Region Servers are the slave nodes. The HMaster gets that request and forward it to the respective Region Server as soon as the client sends a write request to it.

- **Region Server :** It is a system which acts similar to a data node and it receives write request and then directs it to specific Region. Each region has set of row data that can be splited into many column families and it is stored in HStore which consists of Memstore and a set of HFiles. The logs for the read and write operations are kept in Memstore which act similar to name node in Hadoop. Memstore data gets flushed into HFile When certain thresholds are met. HBase buffer sorts data before flushing and writes to HDFS and hence HFile holds a list of sorted rows. The tons of HFiles are created for each frequent Memstore flushes. The request comes from the master it written to WAL (Write Ahead Log) a permanent storage so the data will not lost even when if the data node is down and when data node is up it will perform all the activities and finally everything is flushed out from Memstore.

D. Zoo Keeper:

Zoo keeper instance is started along with the HBase and keep track of all region servers in HBase. Information such as how many region servers are there and which region servers are holding from which data node to which data node one specialty is it keeps track of smaller data sets which Hadoop does not do. It decreases the overhead on top of Hadoop by avoiding maintaining the Meta. The HMaster by actually contacting Zoo keeper it gets the details of region servers.

IV. CONCLUSION

Thus the HBase cloud architecture could able to handle random real-time reading and writing of Big data thus the there is increase in performance than using Hadoop. Utilization of cluster resources is high because of Memstore usage. Thus image processing cloud is able to handle most familiar languages and it also helps to increase the computation speed.

REFERENCES

[1] J. C. Brian Dolan and J. Cohen, "MAD Skills: New Analysis Practices for Big Data," in Very Large DAta Bases(VLDB) 09. Lyon, France: ACM, Aug. 2009.

[2] K. Potisepp, "Large-scale Image Processing Using MapReduce," Mas-ter's thesis, Tartu University, 2013.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In Computer Vision–ECCV 2006, pages 404–417, Springer, 2006.

[4] M.Z. Mosharaf Chowdhury and T. Das, "Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing," in NSDI'12 Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. San Jose, CA: USENIX Asso-ciation Berkeley, Apr. 2012.

[5] C. S. B. Thomas W. Parks, DFT/FFT and Convolution Algorithms: Theory and Implementation. John Wiley & Sons, Inc. NY, USA, 1991.

[6] "Hadoop Introduction," http://hadoop.apache.org/, [Retrieved: January, 2014].

[7] K. K. Muneto Yamamoto, "Parallel Image Database Processing with MapReduce and Performance Evaluation in Pseudo Distributed Mode," International Journal of Electronic Commerce Studies, vol. 3, no. 2, 2012, pp. 211–228. [Online]. Available: http://www.academic-journals. org/ojs2/index.php/ijecs/article/viewFile/1092/124

[8] "Apache CloudStack website," http://cloudstack.apache.org, [

