American Journal of Applied Sciences

Original Research Paper

Processing Large Volume of Biometric Data in the Hadoop Single Cluster Node Environment

Jayakumar Vaithiyashankar, Shohel Sayeed and Anang Muhammad Amin

Faculty of Information Science and Technology, Multimedia University, Malaysia

Article history
Received: 28-02-2017
Revised: 30-05-2017
Accepted: 26-10-2017

Corresponding Author:
Shohel Sayeed
Faculty of Information Science and Technology, Multimedia University, Malaysia
Email: shohel.sayeed@mmu.edu.my

Abstract: In big data evolution, the analysis of large scale data and scrutinizing the required vital information becomes very demanding task. The emerging cloud platform promises and gives hope in handling the enormous volume of data. Hence, a new kind of methodology is required to tap the full potential of leveraging the big data analytics over the biometric data. In this work, we are going to deal with the integration of Hadoop, a map reduce framework with the infamous powerful computer vision library tool, Opencv. The proposed setup will comparatively analyze the large set of biometric data; such as face over the pseudo distributed environment. We test the capacity of our methodology with a different data set and analyze various computational parameters. The results show the proposed method is applicable for dealing in the real distributed environment.

Keywords: Biometrics, Cloud Computing, Distributed Computing, Personal Identification, Face Recognition, Computer Vision

Introduction

The biometric systems tend as legitimate for authentication and identification process at the present time. The biometric storage has surpassed its normal limitation due to upsurge in a large volume of new enrollment across various parts of the world. The face recognition is the most popular among biometric authentication and verification process (Liao *et al*., 2016). Due to the simple nature of data collection, storage and handling face data, it is widely acceptable and recognized all over the globe. Initially the face should be detected from the input image, identified by following some common facial features.

Face identification depends on the image quality, brightness, distortion, noise level and various other factors (Gohil *et al*., 2014). The face should be preprocessed on first hand before doing the recognition phase. After preprocessing lower dimensional representation of the image is obtained and stored. There is a need to extract the features of the low dimensional image using various techniques like Eigen face, Eigen vectors and fisher faces (Karun and Chitharanjan, 2013).

Distributed System

The distributed system is the integration of various computers together to solve particular problem of computing large volume of structured/semi-structured/unstructured data (Shalini *et al*., 2015).

Grid Computing System

The grid computation system is the utilization of the computational resources such as CPU clocks, network storage areas, RAM to process a particular function (Jai and Atul, 2016). The grid computation is the loosely coupled architecture so it returns highly on the investment.

The major problems raised when handling big data are: (1) Runtime (2) Memory Consumption

The runtime stands for entire processing time of the big data sets over the machine (Bruce *et al*., 2016). The memory consumption is the memory required while doing the processing of entire big data can be classified as virtual memory, heap memory and physical memory (Zhuang *et al*., 2015).

The major contribution of this work was integration of the distributed image processing tool in hadoop environment with elastic map reducing algorithm. Depends upon the load request the application is scaled up to the maximum available resource pool.

Design Constraints for Building Biometric Systems

Divergence in Data Formats

There are a large number of biometric data formats available to store. This makes difficult for the conversion of one format to another (Dittrich and Quiané-Ruiz, 2012). There is a huge hindrance in the interoperability due to this diverse nature.

Parallel Computation

The big set of biometric data should be processed simultaneously in parallel fashion to index and sort data as much as fast (Wang *et al*., 2013).

Scalability

The system should be scalable in future depends upon the requirement such that it will be a sustainable model throughout its life-cycle (Anusha and Swetha, 2014).

The Tools used to for Implementation

HDFS

The Hadoop distributed file system was based upon open source implementation of the Google File System (GFS). The various features of the GFS also available in Hadoop as automatic failure management, flexible horizontal scalability, check sum correction and file redundancy (Wu and Hong, 2015). HDFS provides the highest level of fault tolerance over the low cost computer clusters.

The outline of the HDFS working principle as divided into 2 parts as: (1) Map phase and (2) Reduce phase.

Map Phase

In the map phase, the data sets are associated with the key/value pairs and respectively the intermediate results are produced (Qureshi *et al*., 2016).

Reduce Phase

In the reduce phase, It combines all the computed/processed intermediate results with the reference of respective intermediate key/value pairs (Costantini and Nicolussi, 2015).

There are various Hadoop extension flavors available in the market as, Hadoop DB, Hadoop ++, HAIL, Co-Hadoop, ERMS, Trojan HDFS, RCFile, DART, Cheetah, Clydesdale (Chang *et al*., 2015).

HIPI

Hadoop Image Processing Interface, provide the interface for converting the normal images into the Hadoop distributed file system supported format. The HIPI improves the performance and analysis over the image data due to the input split format, hence the big file stored over HDFS can be easily handled (Xu *et al*., 2016).

HIPI supports various image formats like JPEG, PNG, PPM and TIFF. The Hadoop video processing interface makes it suitable for the MPEG format supportable. In Fig. 1, HIPI face database map reduce layout illustrated.

Parallel and Distributed Processing on Hadoop

The architecture of Hadoop is comprised with two major parts, one is HDFS and the second part as MapReduce (Soni *et al*., 2015). The processing can be taken place by two methods via single master server alone or with the help of multiple-slave servers.

HIPI enables the computer vision technology along with Hadoop map reduce. The combination of both Hadoop and HIPI abstracts the higher level of complex technical details involved in integration. It allows the researchers to use computer vision over the Hadoop architecture at most ease.

Hadoop Common

It consists the common library file and default utilities required for the Hadoop basic functions.

HDFS

It allows to access data with very high throughput preferable for data intensive processes.

Hadoop Yarn

It monitors and manages the scheduling of all jobs in Hadoop environment and also tracks the details of cluster resources.

Map Reduce

It is developed based on the principle of YARN, to perform parallel processing over given data. The basic map reduce can be defined as: Each set of job (j) assigned to the key (k) and the value (v). Each key is associated with the value and combined as pairs like Key, value pair $\langle k,v\rangle$.

Mapping Function

The mapping function takes place during the association of key, value pair for all the listed jobs in a cluster.

Mapping function $\langle k^{m,in}, v^{m,in}\rangle$:

$$\begin{aligned}&\left\langle k_1^{m,out}, v_1^{m,out}\right\rangle, \left\langle k_1^{m,out}, v_{(M-1)}^{m,out}\right\rangle, \ldots, \\ &\left\langle k_N^{m,out}, v_2^{m,out}\right\rangle, \left\langle k_N^{m,out}, v_M^{m,out}\right\rangle\end{aligned} \tag{1}$$

Reducing Function

The reducing function takes place after processing the intermediate data and going to combine together with the original key value associated with it:

$$\left\langle k_1^{m,out}, [v_1^{m,out}, \ldots, v_{(M-1)}^{m,out}]\right\rangle \rightarrow k_1^{r,out}, v_1^{r,out} \tag{2}$$

where, m- mapping function, in- input, out- output, r-reducer, M- number of map.

Elastic Map Reduce Function

In the initial step, amount of data is taken into the consideration and depends upon the resources availability the map and reduce task is scheduled over the resource pool. The map and reduce function takes place as per the resource quotient ratio, the lower level leverages to the better distribution of the resources. Thus the computation will takes place in shorter span of time and increases productivity.

The Experimental Setup

The experiment conducted over 3 types of setup environments by varying the operating system with the amount of physical memory gradually. The configuration between 1 and set 2 is only varied with 2 Giga Byte of RAM. Table 1 consist the details of different environmental setups for this experiment.

The details of the face databases used in this experiment given as Table 2. In Fig. 2, software hierarchy for the experiment is illustrated as layer by layer. In Fig. 3, the HDFS architecture is depicted with detailed work flow for better understanding.

Yale Nov (2016) Compromises 165 gray scale data set in GIF standard about 15 subjects. For the each person, different facial expressions are captured by varying: center light, without glasses, happy, sleepy, wink, left light, right light and normal.

Faces 95, Dec (2016), Number of individuals in the database is 72. The Image resolution of every image is 180 by 200 pixels (portrait format) the image database available from computer vision science research projects.

BioID Face DB-Human Scan AG, Switzerland (BioID Oct, 2016), includes 1521 gray scale photos in the resolution size of 384×286 pixels. Every subject is taken for different frontal view of their faces to differentiate from 23 test subjects. Eye positions of each individual is manually set for checking the similarity test.

The Chicago Face Database (Chicago Nov, 2016), created with the intention of scientific research purpose. The database consist uniform photos of both male and female subjects by varying ethnicity, also age ranging from 17 to 65. The Meta data of each photo includes both the physical measurement of individual faces as well as the subjective ranking like attractiveness of each face with the help of separate evaluators.

Georgia Tech face database (Georgia Sep, 2016), encompasses photographs of 50 subjects. All the subject in the data set are categorized with 15 images taken in the scattered background with the resolution of 640×480 pixels. The average size of subject faces in these dataset is about 150×150 pixels resolution.

Table 1. Setup environment with parameters

Parameters	Set 1	Set 2	Set 3
Operating system	Ubuntu 16.04.1 LTS desktop	Ubuntu 16.04.1 LTS desktop	Ubuntu 16.04.1 LTS server
Physical memory	2 GB	4 GB	8 GB
Hard disk size	250 GB	250 GB	500 GB
HIPI version	2.1.0	2.1.0	2.1.0
Open CV version	3.1	3.1	3.1
Motherboard	Intel-pentium-T4400	Intel-pentium-T4400	Intel-i5-2400
Processor speed	2.2 GHz	2.2 GHz	3.10 GHz
JDK version	1.8.0_91	1.8.0_91	1.8.0_91
Hadoop version	2.7	2.7	2.7

Table 2. Face database properties

	Face database		
Database name	Size (MB)	Sample images	Average image size (KB)
Yale faces	6.3	166	35.2
Faces 95	5.9	1440	4.3
BioID-face database-V1.2	167.1	1522	109.8
Chicago face database	1500	1800	857.6
Gergio tech face database	127	750	174.3

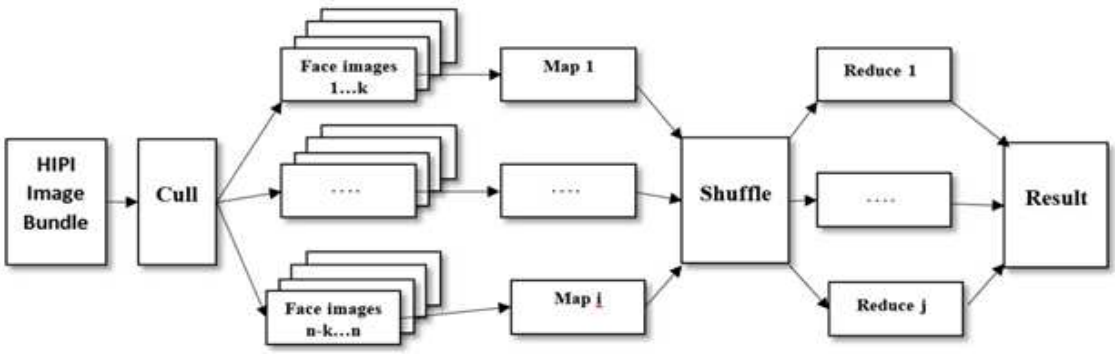

Fig. 1. HIPI face database map reduce layout

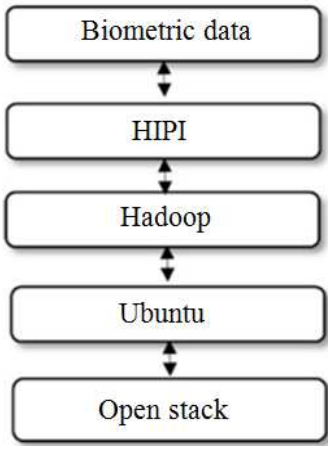

Fig. 2. Software hierarchy stack

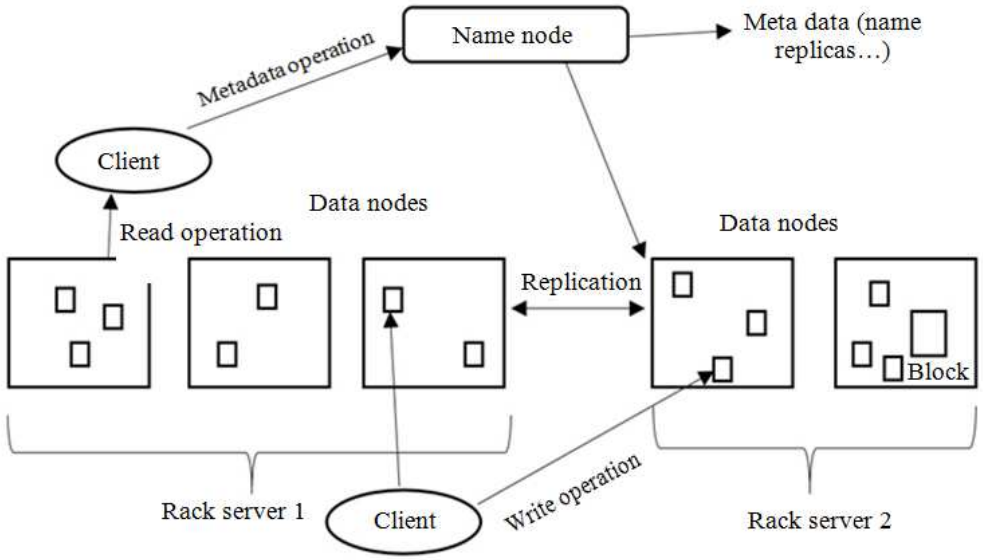

Fig. 3. HDFS architecture

Results and Discussion

Each data set is processed in the predefined 3 different environmental setups individually. The CPU time spent is measured, for the processing and conversion of normal image into the Hadoop supported HIPI image format.

The table shows the comparison between the different data set against various environment.

In the Table 3, the CPU time of database Yale Faces and Faces 95 are negligible, so we compared remaining 3 databases in our comparison chart and plotted as Fig. 4 Comparison of CPU Time.

For example, Bio-ID-face database v1. 2 stands at the set 2 as the minimum value for the computation process against set1 and set 3. Out of total 1521 grey level images resides about the higher resolution as 384×286 pixel depth. Each image is compared against remaining person's facial feature. The training phase records the distinctive facial parameters and features for better recognition in the next phase.

In Fig. 4, the CPU time for bio-id face database v1.2, Chicago face database and Gergio tech face database was compared and plotted as bar chart. The time taken for 1 and set 2 are slightly varied up to 7% overall. Where the time taken for set 3 is only half of the time taken for 1 and set 2. So set 3 computing time is 34% faster than the set 1 and 28% faster than the set 2.

This is due to the Hadoop parallel processing and additional physical memory. Hence the environmental setup gives us the hope to implement in real time multimode cluster environment.

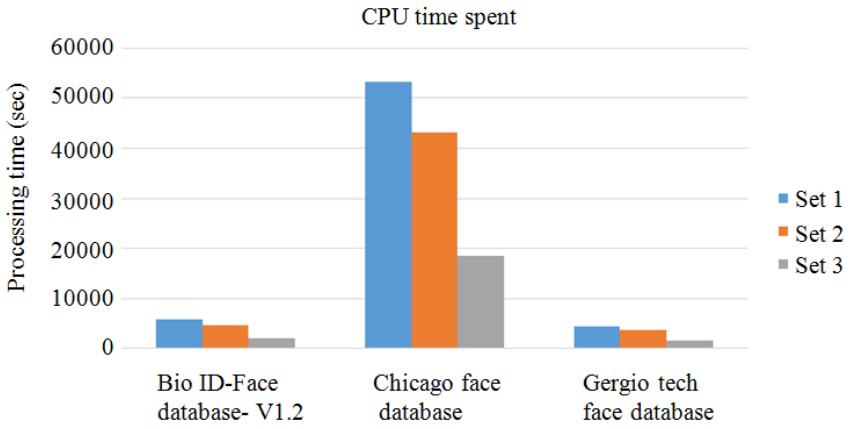

Fig. 4. Comparison of CPU time

Table 3. CPU Time spent

		CPU Time Spent (sec)		
S. No	Database name	Set 1	Set 2	Set 3
1	Yale faces	223.0	180.8	77.5
2	Faces 95	208.9	169.3	72.6
3	Bio ID-face database- V1.2	5915.3	4795.8	2055.3
4	Chicago face database	53170.8	43107.4	18474.6
5	Gergio tech face database	4502.9	3650.6	1564.6

Conclusion

We conclude by providing the holistic view over how biometric data can be integrated with Hadoop environment. In this work, we setup the Hadoop single node cluster over the Ubuntu. The HIPI library tools were installed over the Hadoop environment to efficiently process biometric images. The Open CV was integrated with the HIPI to provide face identification and recognition in this work. The elastic map-reduce function used with these powerful tools produces effective face recognition system at dynamic scale.

Future Work

In the future work, we planned to implement the multi-node cluster in the real server instead of single node environment with multimodal biometric identification.

Acknowledgement

The authors would like to thank Multimedia University, Malaysia.

Funding Information

This work is supported in part by Ministry of Education Malaysia under the FRGS Research grant no: MMUE/140013.

Author's Contributions

The main contribution of this research presented a design of Processing Large Volume of Biometric Data in the Hadoop Single Cluster Node Environment.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of other authors have read and approved the manuscript and no ethical issues involved.

References

Liao, S., A.K. Jain and S.Z. Li, 2016. A fast and accurate unconstrained face detector. IEEE Trans. Pattern Anal. Machine Intell., 38: 211-223. DOI: 10.1109/tpami.2015.2448075

Gohil, P., D. Garg and B. Panchal, 2014. A performance analysis of MapReduce applications on big data in cloud based Hadoop. Proceedings of the International Conference on Information Communication and Embedded Systems, Feb. 27-28, IEEE Xplore press, Chennai, India. DOI: 10.1109/ICICES.2014.7033791

Karun, A.K. and K. Chitharanjan, 2013. A review on hadoop-HDFS infrastructure extensions. Proceedings of the IEEE Conference on Information and Communication Technologies, Apri. 11-12, IEEE Xplore press, Thuckalay. DOI: 10.1109/cict.2013.6558077

Shalini, J., S. Satendra and V. Ashok, 2015. Big data analysis using HDFS, C-MEANS and MapReduce. Int. J. Adv. Res. Comput. Sci. Software Eng.

Jai, V. and P. Atul, 2016. Comparison of mapreduce and spark programming frameworks for big data analytics on HDFS. IJCSC, 7: 80-84.

Bruce, B.R., J.M. Aitken and J. Petke, 2016. Deep parameter optimisation for face detection using the viola-jones algorithm in open CV. Search Based Software Eng. Lecture Notes Comput. Sci. DOI: 10.1007/978-3-319-47106-8_1

Zhuang, H., K. Lu, C. Li, M. Sun and H. Chen *et al*., 2015. Design of a more scalable database system. Proceedings of the 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, May 4-7, IEEE Xplore press, Shenzhen. DOI: 10.1109/ccgrid.2015.70

Dittrich, J. and J. Quiané-Ruiz, 2012. Efficient big data processing in Hadoop MapReduce. Proc. VLDB Endowment, 5: 2014-2015. DOI: 10.14778/2367502.2367562

Wang, L., J. Tao, R. Ranjan, H. Marten and A. Streit *et al*., 2013. G-Hadoop: MapReduce across distributed data centers for data-intensive computing. Future Generation Comput. Syst., 29: 739-750. DOI: 10.1016/j.future.2012.09.001

Anusha, V. and M. Swetha, 2014. High performance clustering on large scale dataset in a multi node environment based on map-reduce and hadoop, Int. J. Emerging Technol. Res., 1: 2347-6079.

Wu, J. and B. Hong, 2015. Multicast-based replication for Hadoop HDFS. Proceedings of the 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, Jun. 1-3, IEEE Xplore press, Takamatsu, Japan. DOI: 10.1109/snpd.2015.7176191

Qureshi, B., Y. Javed, A. Koubâa, M. Sriti and M. Alajlan, 2016. Performance of a low cost hadoop cluster for image analysis in cloud robotics environment. Procedia Comput. Sci., 82: 90-98. DOI: 10.1016/j.procs.2016.04.013

Costantini, L. and R. Nicolussi, 2015. Performances evaluation of a novel hadoop and spark based system of image retrieval for huge collections. Adv. Multimedia, 2015: 1-7. DOI: 10.1155/2015/629783

Chang, B.R., H. Tsai, C. Guo and C. Chen, 2015) Remote cloud data center backup using HBase and Cassandra with user-friendly GUI. Proceedings of the IEEE International Conference on Consumer Electronics, (CCE' 15), Taiwan. DOI: 10.1109/icce-tw.2015.7216976

Xu, W., Y. Shen, N. Bergmann and W. Hu, 2016. Sensor-assisted face recognition system on smart glass via multi-view sparse representation classification. Proceedings of the International Conference on Information Processing in Sensor Networks, Apri. 11-14, IEEE Xplore press, Vienna, Austria. DOI: 10.1109/ipsn.2016.7460721

Soni, S., Y. Wagh and S. Thigale, 2015. Survey paper on hadoop-using a biometric technique "iris recognition". Int. J. Comput. Applic., 114: 11-13. DOI: 10.5120/20005-1944

Yale Nov., 2016. http://vision.ucsd.edu/content/yale-face-database

Faces 95 Dec., 2016. http://cswww.essex.ac.uk/mv/allfaces/

BioID Oct., 2016. https://www.bioid.com/About/BioID-Face-Database

Chicago Nov., 2016. http://faculty.chicagobooth.edu/bernd.wittenbrink/cfd/index.html

Georgia Sep., 2016. http://www.anefian.com/research/face_reco.htm