



Cost based Model for Big Data Processing with Hadoop Architecture

By Mayank Bhushan & Sumit Kumar Yadav

Motilal Nehru National Institute of Technology, India

Abstract- With fast pace growth in technology, we are getting more options for making better and optimized systems. For handling huge amount of data, scalable resources are required. In order to move data for computation, measurable amount of time is taken by the systems. Here comes the technology of Hadoop, which works on distributed file system. In this, huge amount of data is stored in distributed manner for computation. Many racks save data in blocks with characteristic of fault tolerance, having at least three copies of a block. Map Reduce framework use to handle all computation and produce result. Jobtracker and Tasktracker work with MapReduce and processed current as well as historical data that's cost is calculated in this paper.

Keywords: big data, hadoop, cloud computing, mapreduce.

GJCST-C Classification: H.2.8



Strictly as per the compliance and regulations of:



Cost based Model for Big Data Processing with Hadoop Architecture

Mayank Bhushan ^α & Sumit Kumar Yadav ^σ

Abstract- With fast pace growth in technology, we are getting more options for making better and optimized systems. For handling huge amount of data, scalable resources are required. In order to move data for computation, measurable amount of time is taken by the systems. Here comes the technology of Hadoop, which works on distributed file system. In this, huge amount of data is stored in distributed manner for computation. Many racks save data in blocks with characteristic of fault tolerance, having at least three copies of a block. Map Reduce framework use to handle all computation and produce result. Jobtracker and Tasktracker work with MapReduce and processed current as well as historical data that's cost is calculated in this paper.

Keywords: big data, hadoop, cloud computing, mapreduce.

I. INTRODUCTION

Technologies are changing rapidly, with lot of competition. In past, hardware cost was meaningful, as storage was a big issue for technological development, because of it's cost. Software and hardware, both having same cost at that time. After that software becomes complex in terms of development, but easy to use. Nowadays, with decrement in cost of hardware, the limitations of storage is not an issue. As functional programming, works with several functions [1], so it requires large amount of space to run a program, reducing the execution time to a great extent[2]. So today's scenario is about faster execution without focusing on hardware cost. As industry is growing, hardware cost is getting lowered so sufficient amount of storage is available without difficulties. Earlier technologies were having specific views on hardware usage, now even 1TB is not a big deal for our commodity system.

Many social network use Resource Description Framework (RDF) [3]. Facebook's Open Graph [4], Freebase [5] and DBpedia [6] are having structured data. Facebook's Open Graph [4] show connection of user to its real functioning. Freebase [5] provide structured directories for music. DBpedia [6] provide structural contents from wikipedia. As per records till 2012, every minute usage of social networking site

'Facebook', having largest number of users, generating share of 684,478 pieces of contents, 'Youtube' users upload 48 hours video, 'Instagram' users share 3,600 new photos and 'Tumblr' sees 27,778 new post published [7]. A Boeing 737 engine generate 10 terabytes of data in every 30 minutes of flight [8]. All these data are information regarding weather conditions, positioning of plane, travelers information and other matters. So volume, velocity and complexity of data generation is increasing day to day. That require tool to handle it and more importantly with in time limit. Traditional database is not sufficient for doing all these calculation under the time limit. Here Hadoop fulfill all current requirements. Facebook, Google, LinkedIn, Twitter are establishing their business in Big Data. Many companies are still not having Hadoop professionals but they hire those from other companies. World's second largest populated country, India, having four times the population than USA, start trend of Big Data and is implementing Biometric System with unique ID number of every person. This project is called "Aadhar Project" that is world largest Biometric Identity project [9] with use of smart card technology and specification of International standard for electronic identification cards. With research perspective on Big Data, apart from Computer Science, other fields like Mathematics, Engineering, Business and Management, Physics and Astronomy, Social Science, Material Science, Medicine, Arts are also taking keen interest in that [10]. USA is on top, in research of Big Data issues, followed by China [10].

In today's world Big Data is moving towards cloud computing. Cloud computing provides required infrastructure as CPU, bandwidth, storage spaces at needed time. Organization like Facebook, LinkedIn, Twitter, Microsoft, Azure, Rackspace etc. have moved to cloud and doing Big Data analytic work, like Genome Project [11] that is processing petabytes of data in less amount of time. These technologies use MapReduce, for proper functioning. For moving Big Data to cloud, all data is moved and processed at data center [12], as being available at one place, cloud facilities can be easily provided.

In this paper section 2 is focusing on importance of MapReduce technique in current system and its practical uses there. Section 3 elaborate about features of Hadoop system with its functionalities.

*Author α: Department of Computer Science & Engineering, Motilal Nehru National Institute of Technology Allahabad, India.
e-mail: mayankbhushan20061@gmail.com*

*Author σ: Department of Computer Science & Engineering, Indira Gandhi Delhi Technical University; New Delhi, India.
e-mail: sumitarya0072@gmail.com*

Section 4 represent cost optimization while moving data to distributed environment. Section 5 concludes this paper.

II. MAPREDUCE: VISUAL EXPLANATION

MapReduce is framework that work in distributed environment with server and client infrastructure. SPARQL is an RDF query language which used in social networking for data processing. SPARQL produce triplet as result of query process [3]. MapReduce provide functionality for processing query result. Facebook's close friend list, is output of processing of this technology in which 'selection' query processed then 'join' operation start functioning. Every 'join' process run one MapReduce function [13]. This is two layer mapping [3], refer to provide unnecessary MapReduce function for data processing. SPARQL generate triplet form of table in which 'selection' apply followed by 'join' operation. 'Selection' generate KEY-VALUE pair that is need for processing of MapReduce. Triplet ID is KEY assessment while its result is VALUE. Reduce function perform its functionality with same KEY-VALUE pair. 'Multiple join with filter' [3] proposed system with one layer mapping in which filter key used along with 'selection' and 'join' operations.

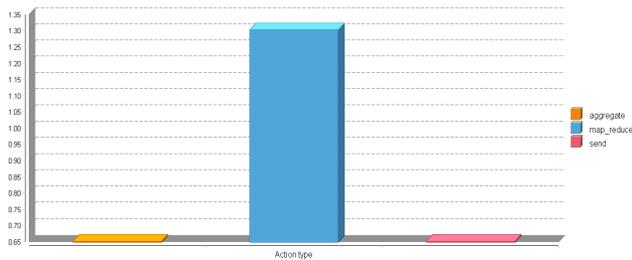


Figure 1 : MapReduce Analysis

Fig. 1 is showing analysis graph of MapReduce function with aggregation of data and sending the data using method of Map and Reduce. Taking an abstract model of Hadoop, MapReduce action is carried out with a rate of 1.65 per unit time, while aggregation and send actions are carried out at a rate of 0.65 per unit time.

MapReduce provides the services as text processing (wordcount, sort, terasort), web searching (pagerank) and machine learning (bayesian classification). HiBench [14] is providing MapReduce function to generate random data to include work load. MapReduce functioning consist four phases as 'map', 'shuffle', 'sort' and 'reduce'. 'Map' process generate intermediate result that need to be process further for resultant, 'reduce' phase start working preceded by shuffle and sort function. If there are 'P' no. of servers in cluster then shuffle phase has traffic $O(P^2)$ flows [15]. The standard concluding output size in Google jobs is 40.3% of the intermediate data set sizes. In the Facebook and Yahoo jobs consider in [16], the fall in

size between the intermediate and the output data is even more distinct: in 81.7% of the Facebook jobs with a reduce segment, the final output data size is only 5.4% of the intermediate data size [15].

Server is responsible for assigning task for MapReduce. If there are 'P' systems and 'N' blocks of data then N/P blocks stored per system by server. Usually block size is user dependent and by default it is 64 MB. 'Map' phase generate (key, value) pair of data where each value have unique ID as key.

Server can run reduce function one time or more. It compute result based on (key, value) pair on server. Task, like web search query reduce function run one time that is sufficient for result [15].

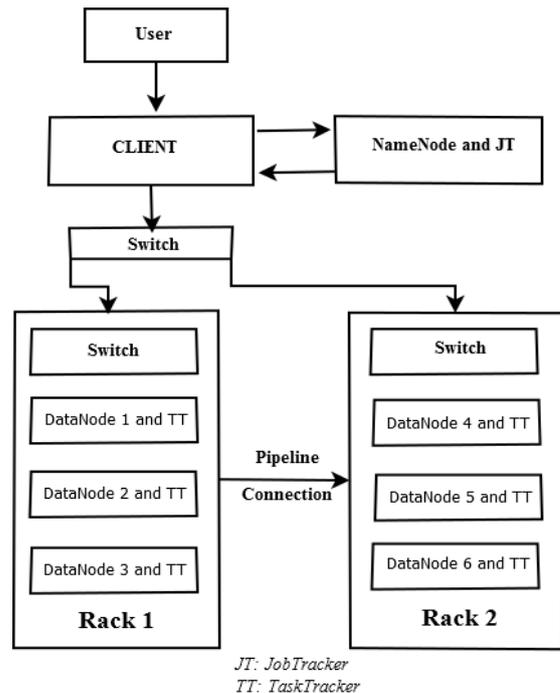


Figure 2 : Architecture of hadoop

They are several presented studies keen on the investigation of MapReduce procedure [17], [18], [19]. Yi Yuan et al. studied MapReduce with bases of CPU utilization, bandwidth, I/O of disk and network usage [20].

III. HADOOP FRAMEWORK SYSTEM MODEL

In recent trends, Hadoop fixing its arms in software industry. Users of traditional database are keen to learn about it. Big Data use Hadoop framework for accessing the data. In 2012, IBM was biggest user of Big Data in revenues followed by HP, Teradata, Oracle, SAP, EMC, Amazon, Microsoft, Google, VMware, Cloudera, Hortonworks, Splunk, 10Gen and MapR [21]. Walmart leading the way with using Big Data on Hadoop for analyzing customer behavior and demand [22]. With huge amount of historical data as match records, individual records, conversations, meeting details etc.,

Australian open start using Big Data for analysis purpose; Netflix is largest commercial video provider in USA, start using Big Data on Hadoop [23]. Here discussion about architecture of Hadoop system with its key feature: Client, Master and slave node and HDFS.

a) *Client*

Client is an application which used by end user and provide task to master and slave node for process. It ensure distributed data processing and distributed data storage. Apart from submitting job to cluster client machine it instruct for 'map' and 'reduce' and at last get the result as output. Client application accept job for processing and break it into blocks. Client take suggestion from master node about empty spaces and distributed these blocks to slaves.

b) *Master Node and Slave Node*

Master node consist with Namenode and Jobtracker while slavenode consist with Datanode and Tasktracker as shown in fig. 2. Client ask Namenode about distribution of blocks. For safety of system block is replicated by minimum three. It is default replicas and it can be set further by user. Namenode provide list of Datanodes to client where data can be stored. Namenode stores meta data which store in RAM that consist information about all Datanodes, racks information, blank spaces, namespace of entire system like last modified time, create time, file size, owner permission, no. of replicas, block-ids and file name. Data retain in Datanode as it never fail; out of three copies one copy retain in by one Datanode in a rack whiletwo other copies put in another same rack but in different Datanodes. This feature gives the quality of fault tolerance with less chance of failure of Datanode and rack simultaneously. Transfer of all block is TCP connection so proper acknowledgment is there with pipeline processing with no wait for completion. Namenode keep updating its meta data as it receives acknowledgment from Datanode. Datanode keep sending signal with interval of three seconds indicating its aliveness; if it not receive by Namenode within 10 minutes then Datanode consider as dead and make it's replica to other node by master node.

If any file need to be executed then client ask Jobtracker to start executing file that reside in Hadoop Distributed File System (HDFS). Jobtracker takes information from Namenode about residence of operative blocks. After that Jobtracker instruct Tasktracker to run program for execution of file. Here 'map' function start and reported by signal to Jobtracker. Output of 'map' result store in Tasktracker's local memory. 'Map' results intermediate data and send it to a node which function by gathering all intermediate data for performing 'reduce' task. At last output is written to HDFS and sent to client.

c) *Hadoop Distributed File System*

Hadoop use HDFS for storing the data that is distributed in nature and storing large data with streaming data pattern. Google file system (GFS) [24] also chunk based file system, use design of one master and many chunkservers. HDFS support fault tolerance with high throughput and can be built out of commodity hardware. But it is not useful for large amount of small files with low latency data access. GFS and HDFS do not execute POSIX semantics [25].

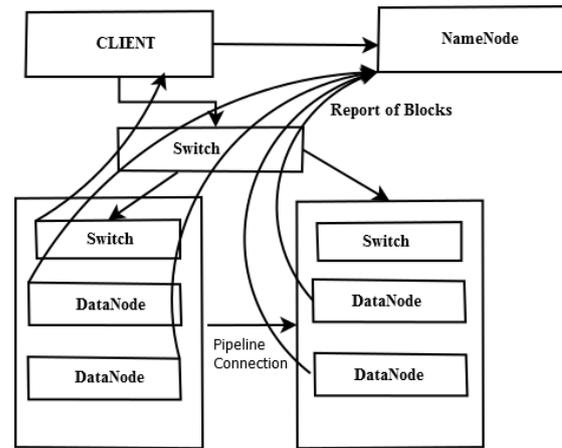


Figure 3 : Connection between Datanode and Namenode

IV. EVALUATION COST IN HADOOP ARCHITECTURE

Consider a system where Client, Namenode, Datanode are connected. Let assume client (C) is connected to switches (P) in client side, Switches (Q) are in Datanode side where (D) numbers of Datanodes are connected to each other in a rack as fig. 3. These racks are connected as pipeline pattern. Such structure is reflected as architecture of Hadoop. Bandwidth between both switches is limited as $B_{P,Q}$

1. When any task comes to client for processing it consult with Namenode. Namenode regularly aware about rack storage for its availability with Datanodes. For engagement of further proceeding value $X_{C,N}$ take decision about connection signal between Namenode and client. Decision cost will be:

$$\text{Decision Cost}(X_{C,N}) = \begin{cases} 1 & \text{if } X > 0 \\ 0 & \text{if } X = 0 \end{cases} \quad (1)$$

2. Client consult with Namenode which have information about rack system. Namenode having knowledge about which Datanode is free to occupy blocks of file which come to client for processing. This file is divided at least in three parts (up to user

choice). Namenode gives the address of maximum bandwidth rack first and continue with decreasing order of bandwidth. If assume data rate is $\eta_{P,Q}$ and total amount need to transfer is $G_d(t)$ then bandwidth cost $B_{P,Q}$ will be:

$$\sum_{t=1}^{t=T} \eta_{P,Q} \left(\sum_{C,p \in P, q \in Q, d \in D} G_d(t) \right) \quad (2)$$

Where p, q, d are one of the component from switches and Datanodes. This information store in RAM of Namenode. Gen2 Hadoop use secondary Namenode which access information for backup of Namenode's data from its RAM and store it to hard disk. Secondary Namenode is not a replacement of Namenode.

3. Datanode store information of current and historical data. As Datanode keep sending signal to Namenode about its aliveness through switch as fig. 3, if Datanode not send signal within 10 min to Namenode then Namenode assume it dead. Storage and estimation cost SSE will be:

$$\sum_{u=1}^{u=t-1} (\gamma) \eta_{P,Q}(u) + \sum_{u=t}^{u=T} \eta_{P,Q}(u) \quad (3)$$

(γ) decide the factor of current or historical data. If any Datanode not sending signal from 10 min. then assume (γ) to 0 but newly allocated data will be transferred to another node by estimation factor.

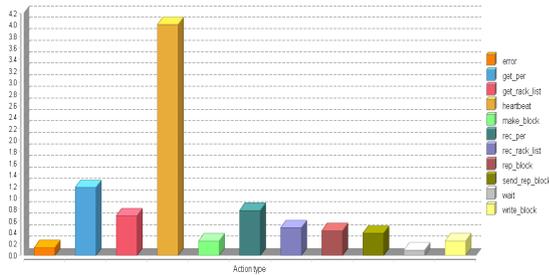


Fig. 4 : Performance Analysis

4. Jobtracker and Tasktracker that are associated with Namenode and Datanode respectively, do MapReduce function. Client load program of Map that executed by Jobtracker for finding situated targeted blocks after consulting by Namenode. Total distribution of blocks are less than number of racks. Tasktracker will produce result that might be 0. Now Reduce task will be executed which collect all intermediate result in a node. That node decide by Namenode and calculate result over there and transfer to HDFS. Resultant data Yred will be:

$$\sum_{u=1}^{u=T} R_{G_d}(u) \quad (4)$$

5. Data move from client to Datanode after generated by user. This data will evaluate routing cost of data which included delay between clien to switches of user side, switches of user side to switches of Datanode side and switches of Datanode side to Datanode. Total routing cost Z_{rt} will be:

$$\sum \zeta(G_d(t))(M_{C,P} + N_{P,Q} + O_{Q,D}) \quad (5)$$

$(M_{C,P} + N_{P,Q} + O_{Q,D})$ is pecuniary cost that showing latency from $C \Rightarrow P \Rightarrow Q \Rightarrow D$. ζ represent constant cost which convert weight cost to monetary cost. (ζ) depend on user as network use.

Fig. 4 is showing overall performance analysis on separate function with action type in X axis and normalized rate in Y axis. First is for error report of decision cost, second is about getting permission between Namenode and client. Third function showing rack list from Namenode. Forth is signal as heart beat which comes on master node in every three second, it is highest time response which happens frequently. Fifth pillar is showing work of client for dividing text file into blocks. Sixth pillar showing receive permission from Namenode, seventh is receiving rack list in which Datanode reside from Namenode so that chunks of file can be allotted. Eighth and Ninth showing replicas information. Data flow in racks as pipeline connection so least waiting rate as showing in tenth pillar. Last eleventh pillar showing action type for writing data into Datanode.

V. CONCLUSION

This paper elaborated the architecture of Hadoop with its growing usage in industries as well as function of MapReduce on which current technologies moving. Among rack that consist of Datanodes and Tasktrackers choose by Namenode on basis of routing cost as showing in paper. It also evaluate cost of result that produce by different Datanodes. Decision of establishing communication of client with Datanode will also be decide by link between Namenode and client. Datanode may consist of historical data that's cost also get evaluated in this paper.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Peter Henderson "Functional Programming Application and Implementation."
2. John Hughes "Why Functional Programming Matters." Institutionen for Datavetenskap, Chalmers Tekniska Hogskola.
3. Liu Liu, Jiangtao Yin, Lixin Gao "Efficient Social Network Data Query Processing on MapReduce." HotPlanet'13 Proceeding of the 5th ACM workshop on Hotplanet, Page(s): 2 - 32.
4. Open Graph, <https://developers.facebook.com/docs/opengraph>.
5. Freebase. <http://www.freebase.com>.
6. DBpedia. <http://www.dbpedia.com>.

7. www.visualnews.com/2012/06/19.
8. www.wipro.com/documents/big-data.pdf
9. www.dataname.com/dataname/2012-08-02
10. Human Genome Project, <http://www.ornl.gov/hgmis/home.shtml>.
11. www.researchtrends.com.
12. Linquan Zhang, Chuan Wu, Zongpeng Li, Chuanxiong Guo, Minghua Chen and Francis C.M.Lau "Moving Big Data to The Cloud." Infocom, 2013 Proceeding IEEE, Turin, 14-19 April 2013, Page(s): 405 - 409.
13. P. Mika and G. Tummarello. "Web semantics in the clouds. Intelligent Systems." Intelligent System, IEEE, 23 Sep.2008, 23(5), Page(s)::8287.
14. S Huang, J. Huang, J. Dai, T. Xie and B. Huang. "The Hibench Benchmark Suite:Characterization of the MapReduce based Data Analysis." 26th International Conference on data Engineering Workshop (ICDEW), 2010, Page(s)::41-51.
15. P. Costa, A. Donnelly, A. Rowtron and G. O'Shea, "Camdoop: Exploiting in-network Aggregation for Big Data application." Proceeding USENIX NSDI 2012.
16. Chen, Y.Ganapathi, A.R.Griffith and Katz R., "The Case for Evaluating MapReduce Performance Using Workload Suite." Modeling, Analysis and Simulation of Computer and Telecommunication System, Singapore, 25-27 July 2011. Page(s)::390-399.
17. A. Pavlo, E. Paulson, A. Rasin, D.J. Abadi, D.J. Dewitt, S. Madden and M. Stonebraker. "A Comparison of Approaches to Large-Scale Data Analysis." Proceeding of the ACM SIGMOD International Conference on Management of Data, NewYork. Page(s):: 165-178.
18. M. Stonebraker, D. Abadi, D.J. Dewitt, S. Madden, E. Paulson, A. Pavlo and A. Rasin. "MapReduce and Parallel dbmss:Friends or Foes?" Communication of the ACM, January 2010. Page(s)::64-71.
19. D. Jiang, B.C. Ooi, L.Shi and S.Wu. "The Performance of MapReduce: An in-depth study." Proceeding of the VLDB Endowment. 1-2 September 2010. Page(s):: 472-483.
20. Yi Yuan, Haiyang Wang, Dan Wang, Jiangchuan Liu. "On Interference-aware Provisioning for Cloud-based Big Dat Processing." 21st International Symposium on Quality of Services. 3-4 June 2013. page(s)::1-6.
21. <http://www.networkworld.com/slideshow/114134/>.
22. <http://www.gigaom.com/2011/07/17>.
23. <http://www.computerworld.com/slideshow>.
24. S. Ghemawat, H. Gobioff and S. Leung. "The Google File System." SIGOPS Operating System Review. Page(s)::29-43.
25. Changqing Ji, Yu Li, Wenming Qiu, Uchechukwu Awada, Kequie Li. "Big Data Processing in Cloud Computing Environments." 12th International Symposium on Pervasive System, Algorithm and Networks, San Marcos. 13-15 Dec. 2012. page (s) ::17-23.

This page is intentionally left blank