Minimizing Network traffic on cloud environment with MapReduce technique

A .Sai Swaroop[1], G.C Sathish[2]
[1]*PG. Student, School of Computer Science Engineering, Reva University, Bangalore.*
[2]*Professor, School of Computing and Information Technology, Reva University, Bangalore.*

Abstract - Map Reduce is the backbone of Hadoop, providing the versatility and simplicity data processing required for a huge quantity of processing data and storage. Map Reduce is procedure that involves two phases, the card phase and the reduction phase. For MapReduce the input comes from Hadoop Distributed File System (HDFS). Mapping tasks here require the input of a huge quantity of data or information and convert it into another form of data, represented as pairs, called key-value pairs. Then follow the Reduce task whose entry is the new representation of data per card task. Here a collection of parallel reduction tasks is executed. Conventionally, a Name node is used in Map Reduce to keep records of the separation of data, the structure of the file, and how data fragments are placed. At startup, the data node sends the entire list of fragments to the name node. When the data file changes, new number is assigned. Because the new version may have changes that differ only from the previous version, it causes a huge quantity of replicated data. To upgrade technique of Map & reduce jobs, many proposed systems have been configured, but they are not monitoring and reducing the network load. Here we suggest and simulate a method to reduce traffic by diminishing group of duplicate or replicated files in the cloud. This is suggested and simulated with two main concepts. The first is a file redirection. This happens in the case when a client requests a server and the requested server is busy, then the request is redirected to another server. The second idea is replication control. This concept is described using a logical block addressing algorithm (ie, LBA) and MD5 (Message Digest 5), which generates a unique hash code for the contents in the files, hence avoiding same data from being replicated.

Keywords - *Hadoop, MapReduce, Big Data, Network traffic.*

I. INTRODUCTION

The simple model of programming and the automatically managed parallel execution has made MapReduce a most important framework for computing of the BigData. The top companies such as Facebook, Google and Yahoo adopt MapReduce as it is an implementation of Hadoop which is open source and is very useful for various applications of big data like cyber security, machine learning etc. The map and Reduce are the two main phases in which the MapReduce has been divided into, these two part will carry out numerous map and reduce tasks. The original input splits are parallel converted into intermediate data which is in the form of value/key pair which is taken care of by the map phase. This key/value pair is later stored in one of the local machine and is arranged in multiple partitions of data. The final result is generated by the reduce task by fetching its own set of data from each of the partition in which it is stored and generates the final result. Between every map and the reduce phase there is a phase known as the shuffle step. In the shuffle step the result of the map phase which is the data which is produced are partitioned , ordered and then transferred into the machines in which the reduce phase is going to be executed. The traffic from network results in a pattern from the tasks conducted by the map and transfers it to the reduce task which causes a large volume of network traffic and imposes a constraint which affects the efficiency of applications of the data analytics.

II. RELATED WORK

[1] Processing and generation of huge sets of data is done by utilizing an implementation and a programming model known as MapReduce. A map function can be generated which uses a key and a value pair. This is then proceeded by generating a group of intermediate key/value pairs. A reduce function is also required which merges all the values which are linked to the same key . As per Jeffrey Dean and Sanjay Ghemawat, the various details of the partition of data input and the scheduling execution of the program, handling of the machine failure and communication among the inter-machine is handled at the run-time by the system. the most frightening resources if the Network Bandwidth, and therefore the optimization of the locality is used, which is nothing but reading data and wring a unique copy of the data from locally stored disks or drives and save it to the network bandwidth.

[2]Weina Wang, Lei Ying, Kai Zhu, Li Zhang and Jian Tan proposed in their paper an algorithm which includes Join the Shortest Queue Policy and Max-Weight Policy, these policies achieve a full capacity of region and minimizes the backlogged tasks . Throughput optimal through the simulations and the heavy freight was proved in this algorithm .

[3]MapReduce consists of 3 phases: Map, Shuffle and Reduce phase. Fangfei Chen, et al. implemented an algorithm that reduced the weighted time of response.

[4]The files are replicated and usual kept on different computers in a distributed response. due to this files replication it consumes a large storage space and wherever possible it becomes crucial to regain those space. As per the observation half of the consumed space was consumed by these duplicate and replicated file. Dan Simon, Marvin Theimer, John R. Douceur, William J. Bolosky and Atul Adya designed a framework which reclaims the spaces from his duplicate data and it consisted of convergent encryption and SALAD [4]. In spite of the files being encrypted by the various keys, the

convergent encryption give a way for the replicated files to come together and merge into a single file, SALAD is also known as Self-Arranging, Lossy, Associative Database and it is a name given for a database which integrates the contents of the file and the various information about the fault tolerant, scalability and decentralized manner in a placement

[5] A problem which proves the data integrity of the data stored on untrusted device has come into demand because of the resource sharing and outsourcing services. I the Provable Data Procession(PDP) model some amount of data is kept by performing processing and then storing it on the untrusted server

III. PROBLEM STATEMENT

The data generated by the map phase is ordered, partitioned and transferred to the appropriate machines that execute the reduction phase. The pattern of network traffic resulting from all card tasks to all redundant tasks can result in a large amount of network traffic, which greatly limits the efficiency of data analysis applications.

IV. SYSTEM ARCHITECTURE

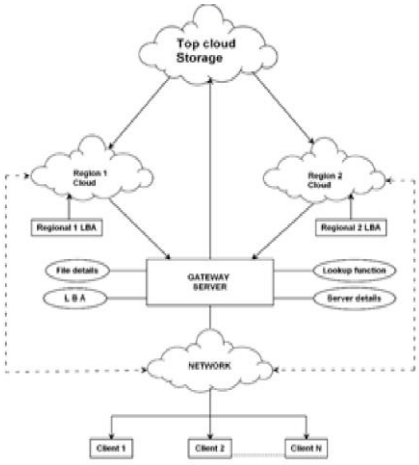

Fig 1.1 System Architecture

Key features illustrated in the architecture Fig 1.1 are:

- LBA (Logical Block Addressing) in the system is a scheme that is used for keeping information about location of storage of blocks of data that are formed from files.
- Regional 1 LBA and Regional 2 LBA are the two sub regions that are differentiated by the IP addresses that they get requests from.
- Look up Function: Here in IP look up details file names and its associated hash tag for data are stored and matched for every new file upload or download request.

The proposed framework has two modules. These modules are explained in brief as follows. There are two modules

- Admin Module: As the name suggests, admin is the administrator who has all rights. Admin account is the type of super account that has unique functionalities. Admin account has the privilege to view all clients' details, hash details and cloud details. Hash details refers to the hash codes generated for the files that are uploaded. Admin account also has the option to view transactions of all clients. Admin can add, edit or delete users.
- User Module : Client is referred to the end users, one who can upload or request to download files. Client can view and edit his details. One can upload file or request to download a file. Uploading a file leads to the generation of hash code for the data in the file. A client can view his transactions.
-

V. IMPLEMENTATION

Map Reduce is a main feature of Hadoop which gives the necessary scalability and easily processes the data for large volume of storage and processing of data. Map and Reduce are the two phases of MapReduce. Hadoop Distributed File System(HDFC0 provides the input for the MapReduce. The large volume of data is provided as an input for the map which transforms it into a different form of data known as the Keys/Value pair. The new data which is the output of the map phase is sent as an input to the reduce phase. in this step a collection of reduce tasks are performed parallely. This makes use of the MVC architecture. The web application is built using a framework which is called the Model View Controller. The logical part of the application data is handled as a main part of the Model. The data display portion is handled by the view part of the application. The interaction among the users is handled by the Controller part of the application.

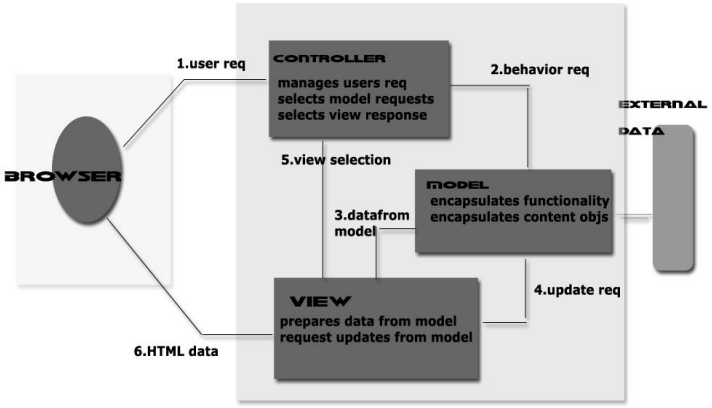

Fig. 1.2 Data Request and retrieval process

A. *Admin Module*

a) Admin Login

Administrator ought to give administrator id and secret key to login.

b) Show Profile

Admin can see his profile details and he can edit profile details and password.

c) User Details

Administrator can see all user subtle elements. He can include new user, alter existing user details, ready to erase any user account.

d) Cloud Details

Admin can see all cloud account details.

e) Hash Details

Admin can see the hash tag generated for all blocks which is uploaded by users.

f) Transactions

Admin can see transaction details of any user by selecting the desired user account.

g) Sign Out

After finishing the current task Admin can logout by clicking on sign out option to come out from ongoing session.

B. User Module

a) User Login

User should provide user id and password for login.

b) Show profile

User can see his profile details and he can edit profile details and password.

c) Upload File

- User has to select the file from system and click to upload option.
- Files will get divided into small blocks (500 bytes each block).
- Generate hash tag for all block.
- Compare generated hash block with existing hash tag from database if hash tag matched in that case we will not upload that block into cloud, we will increase number of instance of that block in database table.
- If hash tag not matched in that case we will add that block hash details in database and upload that block in cloud.
- LBA - Logical Block Addressing technique is used to identify what are the blocks are present in a file.

d) Download File

- User has to select a file to download.
- Request Send to the Server, Server has to fetch the IP address of the client system and using Look-UP function identify the IP address Region, let it be Selected Region (SR).
- Each Region has one Storage Space and One Map Area. In Storage space all the downloaded blocks data are available and in Map area downloaded block details are available.
- Using LBA server has to find block numbers which are in selected file.
- Server has to check in Map area of SR whether all the blocks required for the file is available, if all the blocks are available in SR storage space download and merge the blocks and give it to the user. If few blocks are available and few are missing then get the missing blocks from Top Region and place it in SR storage space and update the SR Map details.
- Transactions
- User can see all its transaction details.
- Sign Out
- After finishing the all task, user should click on sign out option to come out from existing session.

VI. CONCLUSIONS

We study the joint optimization of intermediate data partition and aggregation in MapReduce to minimize network traffic cost for big data applications. We propose a three-layer model for this problem and formulate it as a mixed-integer nonlinear problem, which is then transferred into a linear form that can be solved by mathematical tools. To deal with the large-scale formulation due to big data, we design a distributed algorithm to solve the problem on multiple machines. Furthermore, we extend our algorithm to handle the MapReduce job in an online manner when some system parameters are not given.

It concludes that the network traffic can be reduced in both offline and online cases. This is observed by the proposed methodology where there is an optimization of redundant files in the storage. This is achieved by the IP look up concepts where we are doing a file redirect to reduce the network traffic when requested cloud region is busy. Also when the files are saving multiple times by the different servers, the replication control concept starts implementing where replicated multiple files cannot be saved in the cloud which is achieved by the MD5 the hash code generation.

VII. REFERENCES

[1]. Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113.

[2]. Wang, W., Zhu, K., Ying, L., Tan, J., & Zhang, L. (2016). Maptask scheduling in mapreduce with data locality: Throughput and heavy-traffic optimality. IEEE/ACM Transactions on Networking, 24(1), 190-203.

[3]. Chen, F., Kodialam, M., & Lakshman, T. V. (2012, March). Joint scheduling of processing and shuffle phases in mapreduce systems. In INFOCOM, 2012 Proceedings IEEE (pp. 1143-1151). IEEE.

[4]. Douceur, J. R., Adya, A., Bolosky, W. J., Simon, P., & Theimer, M. (2002). Reclaiming space from duplicate files in a serverless distributed file system. In Distributed Computing Systems, 2002.

[5]. Proceedings. 22nd International Conference on (pp. 617-624). IEEE. Erway, C. C., Küpçü, A., Papamanthou, C., & Tamassia, R. (2015).

[6]. Dynamic provable data possession. ACM Transactions on Information and System Security (TISSEC), 17(4), 15.

[7]. Diffie, W., and Hellman, M.E., New Directions in Cryptography, IEEE Transactions on Information Theory, vol. 22, no. 6, November 1976, pp.

[8]. Garret, Paul. Making, Breaking Codes: An Introduction to Cryptology. Upper Saddle River, NJ: Prentice-Hall, 2001.

[9]. Kurose, James F., Ross, Keith W., Computer Networking: A top Down Approach Featuring the Internet. 2nd edition. Addison Wesley 2002.

[10]. J. Lenstra, A. Rinnooy Kan, and P. Brucker, "Complexity of machine scheduling problems," Annals of Discrete Mathematics, vol. 1, pp. 343–362, 1977.

[11]. J. Lenstra and A. Kan, "Complexity of scheduling under precedence constraints," Operations Research, pp. 22–35, 1978.

[12]. P. Brucker, Scheduling algorithms. Springer, 2004.

[13]. Q. Xie and Y. Lu, "Degree-guided map-reduce task assignment with data locality constraint," in Proc. IEEE ISIT, Cambridge, MA, USA,2012, pp. 985–989.

[14]. J. Tan, X. Meng, and L. Zhang, "Coupling task progress for MapReduce resource-aware scheduling," in Proc. IEEE INFOCOM, Turin, Italy, Apr. 2013, pp. 1618–1626.
[15]. M. Isard et al., "Quincy: Fair scheduling for distributed computing clusters," in Proc. ACM SOSP, Big Sky, MT, USA, 2009, pp. 261–276.
[16]. Rajesh, M., and J. M. Gnanasekar. "Congestion control in heterogeneous WANET using FRCC." Journal of Chemical and Pharmaceutical Sciences ISSN 974 (2015): 2115.
[17]. Rajesh, M., and J. M. Gnanasekar. "A systematic review of congestion control in ad hoc network." International Journal of Engineering Inventions 3.11 (2014): 52-56.
[18]. Rajesh, M., and J. M. Gnanasekar. " Annoyed Realm Outlook Taxonomy Using Twin Transfer Learning." International Journal of Pure and Applied Mathematics 116.21 (2017) 547-558.
[19]. Rajesh, M., and J. M. Gnanasekar. " Get-Up-And-Go Efficientmemetic Algorithm Based Amalgam Routing Protocol." International Journal of Pure and Applied Mathematics 116.21 (2017) 537-547.
[20]. Rajesh, M., and J. M. Gnanasekar. " Congestion Control Scheme for Heterogeneous Wireless Ad Hoc Networks Using Self-Adjust Hybrid Model." International Journal of Pure and Applied Mathematics 116.21 (2017) 537-547.