KIET IJCE KIET International Journal of Communications & Electronics
Volume. No. 5, Issue No. 1, Jan-June 2017, ISSN: 2320 – 8996

S-Hadoop -An enhancement of H2Hadoop

Shruti Satpute[1], Shalaka Zaware[2], Ganesh Phapale[3], Shubham Shahane[4]

[1,2,3,4]*Department of Computer Engineering,*

Amrutvahini College of Engineering, Sangamner, Ahmednagar,Maharashtra,India

SavitribaiPhule Pune University, Maharashtra, India

[1]shrutisatpute7@gmail.com

[2]shalakazaware@gmail.com

[3]ganeshphapale77@gmail.com

[4]32shubham32@gmail.com

***Abstract*-Cloud Computing supports Hadoop framework to process BigData in parallel. Hadoop has certain limitations that could be exploited to execute the job efficiently. These limitations are mostly because of data locality in the cluster, jobs and tasks scheduling, and resource allocations in Hadoop. Maintaining efficiency in resource allocation still remains a challenge in Cloud Computing. MapReduce platforms. H2Hadoop is an enhanced Hadoop architecture that reduces the computation cost associated with BigData analysis. It addresses the issue of resource allocation in native Hadoop,also it provides an efficient. Data Mining approach for Cloud Computing environments. H2Hadoop architecture leverages on NameNodes ability to assign jobs to the TaskTrackers (DataNodes) within the cluster. By adding control features to the NameNode, H2Hadoop can intel- ligently direct and assign tasks to the DataNodes that contain the required data without sending the job to the whole cluster but the single node gets overloaded due to this. A S-Hadoop system is proposed which makes use of concepts like data deduplication and indexing. S-Hadoop uses efficient searching algorithms which will solve this problem. An efficient searching technique for retrieval of data is been proposed.**

***Keywords*— Cloud Computing, BigData, Hadoop, H2Hadoop, S-Hadoop, Fault tolerance, HDFS, Name node, Data node, MapReduce, Text Data, Data Deduplication, Indexing.**

I. INTRODUCTION

ThisHadoop is an Apache project, Java based open source framework which distributes processes of huge database sets through clusters of computers using models of programming. A Hadoop works in an background frame work that provides distributed storage and computing system across computers cluster. Hadoop is designed to increase scalability from one server to multiple of machines.[1]

H2Hadoop is an enhanced Hadoop architecture that reduces the computation cost associated with BigData analysis. It addresses the issue of resource allocation in native Hadoop,also it provides an efficient Data Mining approach for Cloud Computing environments. H2Hadoop architecture leverages on NameNodes ability to assign jobs to the TaskTrackers (DataNodes) within the cluster. By adding control features to the

NameNode, H2Hadoop can intelligently direct and assign tasks to the DataNodes that contain the required data without sending the job to the whole cluster but the single node gets overloaded due to this. In this project an enhanced version of H2Hadoop is proposed. Which is S-Hadoop system.A S-Hadoop system is proposed which makes use of concepts like data deduplication and indexing. S-Hadoop uses efficient searching algorithms which will solve this problem. An efficient searching technique for retrieval of data is been proposed.

Use of Indexing and Deduplication is done to perform further operations on BigData.

II. EXISTING SYSTEM

A. HadoopOverview

Hadoop is HDFS In which a MapReduce process that application is divided into many small pieces of task each of which may be executed on the nodes of Hadoop Clusters. MapReduce provides flow reading access, runs tasks on a cluster of nodes, and provides a data controling system for a expanded data storage system. Application for transferring data to the databases Mysql and Hadoop. It can make compatible with incremental loads of a single table or a free form SQL query MapReduce algorithm has used for applications like creating search indexes, document clustering, access logs, and different data analysis. “one time write and multiple times read” this can be permites to data files to be written only once in HDFS and then permites to be multiple times reading with respect to the numbers of distributed jobs. During the writing process, Hadoop distributes the information into predefine blocks with size.[1]

B. Common Job Block Table

The H2Hadoop MapReduce workflow is the same as the original Hadoop in terms of hardware, network, and nodes. However, the software level has been enhanced. We only add features in NameNode that allow it to save specific data in a look up table which is called as Common Job Blocks Table CJBT.

C. The Common Feature

Common Features are define as the shared data between jobs. H2Hadoop supports caching, enables output to be written in the CJBT during the reduce step. We use Common Features to identify the DataNodes or the blocks with shared data entries.

III. PROPOSED SYSTEM

In the given paper, system of data searching and avoiding the complexity we implement the concept of the paper in which by using the MySQL, BigDataHadoop and using Java Language project implemented. In the proposed system firstly run Hadoop on the windows 7 operating system by using virtual machine(VMware). After the Hadoop execution creating the cluster and node to HDFS. Then Mapreducing can be done for sorting the input. In this project building a cluster for the proposed solution following some directions to prepare the cluster first, then we can do the modifications on the environment. In addition, since in this have Hadoop and MySQL both run on a shell interface of windows, in this will use it for the implementation of the proposed solution.

A. *MapReduce*

MapReduce is a framework which is used for easily distribution, fault-tolerant manner. A MapReduce job usually splits the input set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. The execute nodes and the storage nodes are the same, MapReduce and the HDFS are running on the same set of nodes The MapReduce framework consists of a one master and single slave TaskTracker for each cluster-node.[12]

B. *S-Hadoop*

No Parallel processing in Cloud Computing has emerged as an interdisciplinary research area due to the heterogeneous nature and large size of data. Sequential pattern mining or data analysis applications such as reduce the amounts of data processing and computational capabilities.[9] Efficiently targeting and scheduling of computational resources is required to solve such complex problems. In Hadoop architecture location of the blocks in HDFS is knowing by NameNode. NameNode is allocate the tasks to a client and divides that job into tasks. It knows which DataNode holds the blocks containing the optional data. NameNode able to direct the jobs to the specific DataNodes without going through the whole cluster. In H2Hadoop, first we add a pre-processing phase in the NameNode and then assigns the tasks. In which focus is on identifying and extracting features to build a metadata table that store information of the location of the data blocks with job name and features.

1) *Flowchart:*

Flowchart explained in Figure 1, we can see that there are two more conditions i.e Indexing and Datadeduplication in S-Hadoop. when compared with H2Hadoop that perform with a delay in job processing. However, if we have a relationship between jobs,S-Hadoop performance will be better than the H2 Hadoop.[11]

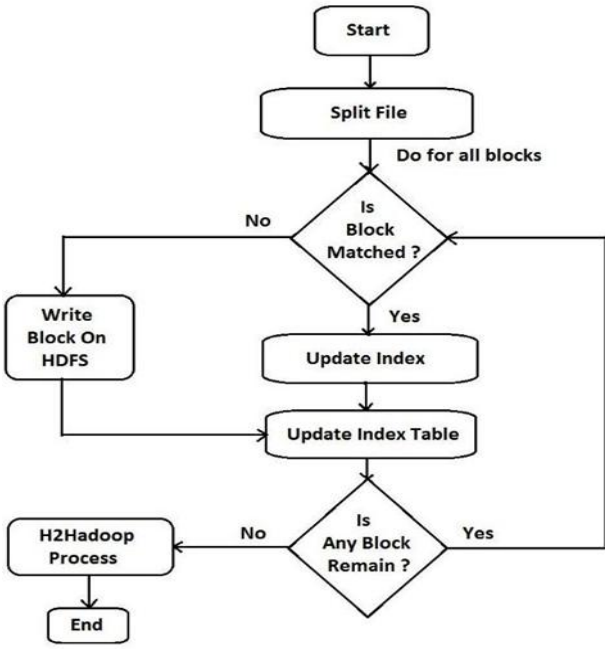

Fig. 1 S-HadoopFlowdigram

2) *Algorithms:*

Indexing Algorithm:

Step 1: Take multiple databases(two or more)
Step 2: Sort the database using BKV"s(blocking key value). Fix the window size w and it should be always greater than 1 i.e.(w>1) This window is moved over data sequentially to get similar record pairs.
Step 3: Linking of records is done.
For record linkage consider following terms,
(nA + nB): total no of records in both the databases
(nA + nB − w + 1): total no of window positions.
Total no of unique candidate pairs generated equals to:
uSNDSA = w(w − 1)/2 + (nA − w)(w − 1)
= (w-1)(nA-w/2)

Pseudo Code:

Step 1: Start
Step 2: Take the multiple databases (two or more)
Step 3: Sort the databases using BKV's (Blocking key values)

a. Fix the window size for comparison, (w>1), which will move sequentially over data.
b. Compare record pairs within current window.

Step 4: BKV's are inserted into array that is sorted alphabetically from left hand side of array

Step 5: Then window is moved over this array and candidate record pair are generated in current window.
Step 6: In case of record linkage, BKV"s from both databases will be inserted into combine array and sorted alphabetically, such that for each pair one record is selected from each of two database.
Step 7: End

3) *Architecture Workflow:*

A S-Hadoop Architecture has been explained in figure 2 as follows:

Step 1: Client " A" sends a request to NameNode. The request includes the need to copy the data files to DataNodes.

Step 2: NameNode replays with the IP address of DataNodes. In the above diagram NameNode replies with the IP address of five nodes (DN1 to DN5).

Step 3: Client " A" accesses the raw data for manipulation in Hadoop.

Step 4: Client "A" formats the raw data into HDFS format and divides blocks based on the data size. Update index after checking DB. In the above example the blocks B1to B4 are distributed among the DataNodes.

Step 5: Client "A" sends the three copies of each data block to different DataNodes.

Step 6: In this step, client "A" sends a MapReduce job (job1) to the JobTracker daemon with the source data file name(s). Step

Step 7: JobTracker sends the tasks to all TaskTrackers holding the blocks of the data.

Step 8: Each TaskTracker executes a specific task on each block and sends the results back to the JobTracker .

Step 9: JobTracker sends the final result to Client "A". If client "A" has another job that requires the same datasets it repeats the set 6-8.
Step10: In native Hadoop client "B" with a new MapReduce job (job2) will go through step 1-5 even if the datasets are already available in HDFS. However, if client "B" knows that the data exists in HDFS, it will send job2 directly to JobTracker.
Step 11: JobTracker sends job2 to all TaskTrackers.
Step12: TaskTrackers execute the tasks and send the results back to the JobTracker.
Step 13: JobTracker sends the final result to Client "B". [1]

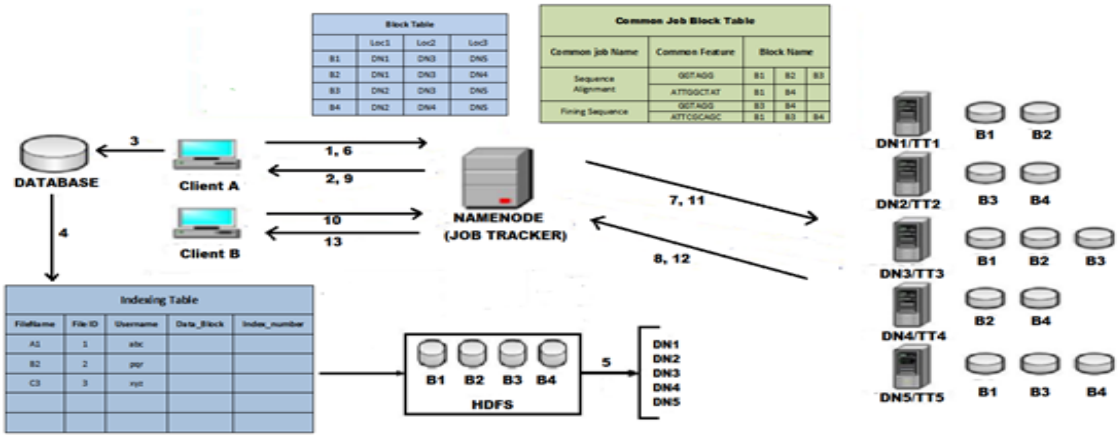

Fig. 2 S-Hadoop Architecture

C. *Indexing*

In H2Hadoop, by design – massively parallel, scalable, data agnostic, and hardwired to full row or column scans is not suitable for indexing. This essential technology is key to traditional DBMS, keeping the performance high-ground over h2Hadoop on (near) real-time analytical tasks. But using indexing algorithms and data deduplication techniques we present enhanced version of H2hadoop that is S-Hadoop.

S-Hadoop Indexing table , addressing the indexing problem at the beginning of the S-Hadoop pipeline when data is uploaded to HDFS. In Indexing table FileName, fileID, User_name, data_block, index_number parameters are used. S-Hadoop, by default, keeps three identical copies of each data block distributed between datanodes in the cluster for redundancy and parallel performance through data locality. The indexing achieved a speed-up of up to 20 times by mainly reducing the data required to be read by mappers. This requires anticipation of the workload for optimal index design, and extra time to create the indices. It also update information parallely.

IV. Conclusions

In Enhanced H2Hadoop framework (S-Hadoop), which allows a NameNode to identify the blocks in the cluster where certain information is stored. And updating the index. A secure, efficient scheme is proposed, which searches common blocks in Bigdata and allows us to perform operation on it which results in less memory storage, CPU time cost. In turn giving faster access and analysis of data.

Acknowledgment

Itgivesusgreatpleasureinpresentingthe paper on S-HADOOP ANENHANCEDH2HADOOP. We would like to take this opportunity to thank our guide Prof. S. A. Thanekar for giving us all the help and guidance we needed. We are really grateful to him for his kind support. His valuable suggestions were very helpful. And Also thanks to Michael Shell and other contributors for developing and maintaining the KIETIJCELaTeX style files which have been used in the preparation of this template.

References

[1] HamoudAlshammari, Jeongkyu Lee and Hassan Bajwa,H2Hadoop: Improving Hadoop Performance using the Metadata of Related Jobs,IEEE TRANSACTIONS ON Cloud Computing, manuscript ID TCC-2015-11-0399.

[2] HamoudAlshammari,JeongkyuLee,HassanBajwa: Improving Current HadoopMapReduce Workflow and Performance", Volume 116 No. 15, April 2015.

[3] Abhilasha Singh, ChetaliParve, PriyankaTripathi'Analysis of Map Reduce Performance using Prefetching Mechanism", Volume 128 No.11, October 2015.

[4] ZebaKhanam and ShafaliAgarwal Map-reduce implementation performance comparison",August2015.A. Ahmad Sharif, B. AkramNoorollahi Intrusion Detection and Prevention Systems (IDPS) and Security Issues ", International Journal of Computer Science and Network Security, VOL.14 No.11, November 2014

[5] Qi, C., L. Cheng, and X. Zhen, Improving MapReduce Performance Using Smart Speculative Execution Strategy. Computers, IEEE Transactions on, 2014. 63(4): p. 954-967..

[6] Farrahi, K. and D. Gatica-Perez, A probabilistic approach to mining mobile phone data sequences. Personal Ubiquitous Comput., 2014. 18(1): p. 223-238.

[7] A. Ahmad Sharif, B. AkramNoorollahi Intrusion Detection and Prevention Systems (IDPS) and Security Issues ", International Journal of Computer Science and Network Security, VOL.14 No.11, November 2014

[8] RongGua,XiaoliangYanga,JinshuangYana,YuanhaoSunb,BingWangb, ChunfengYuana, YihuaHuanga'SHadoop: Improving MapReduce performance by optimizing job execution mechanism in Hadoop clusters", 2013.

[9] Ming, M., G. Jing, and C. Jun-jie. ,Blast-Parallel: The parallelizing implementation of sequence alignment algorithms based on Hadoop platform. in Biomedical Engineering and Informatics (BMEI), 2013 6th International Conference on. 2013.

[10] Jian Tan, XiaoqiaoMeng, Li Zhang: Coupling Task Progress for MapReduce Resource Aware Scheduling", 2013.

[11] Wei Luo, Nicholas Woodward for Analysis and Optimization of Data Import with Hadoop, 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops.

Matsunaga, A., M. Tsugawa, and J. Fortes. CloudBLAST: Combining MapReduce and Virtualization on Distributed Resources for Bioinformatics Applications. ineScience, 2008. eScience '08. IEEE Fourth International Conference on. 2008..