Hadoop Cluster Performance with MapReduce and Pig Latin in the Big Data

K.Umapavan Kumar, S. V. N Srinivasu, A. Ramaswamy Reddy

Abstract*: *The huge amount of data population in the current scenario incurring two major issues one is storage and other is processing of the data. The big data scenarios like social media, search engines and other applications generating humongous data which need be separately handled when compared with existing storage and processing techniques. The important point is the way of storing the data and processing the data. The current discussion addressing the Hadoop framework internals and the capability of the Hadoop cluster along with the processing of map reduce and pig Latin scripts. The main goal is to analyze the environment of map reduce and pig scripts with a method of estimating the factors like time and space requirements along with the input splits and output splits in a detailed manner. The existing works in Hadoop internals not focused much on these aspects and sure the discussion creates a road map to study the architectural aspects which can be helpful to the researchers to enhance the existing architectures in a better possible way. On the other hand adopt the new techniques like analytics and Machine Learning libraries based on the requirements of the industry. The reason behind this work is to pin point the usage of map reduce and the complementary aspects like pig and summary of the various parameters to suggest usage path to the developers. The work also provides some analytics to conclude the suitability of the application running in the context of Map Reduce and Pig Latin

Index Terms*: *Big Data, MapReduce, Hadoop Pig, unstructured.

I. INTRODUCTION

Big data scenarios are everywhere in the world to name a few examples we have social media data population in the form of likes, shares and comments [1-3]. The additional things performed by social media applications are bit interesting. For example, in Facebook the most viewed picture of a profile and friend suggestions and remainder posts like lost year events and comments related to a particular post. The flow of the work is as follows in section I various scenarios of the big data the need of huge collection of the data and parallel processing is needed. Section II describes the usage of Map Reduce frame work with experimental setup and the estimation of required amount of memory and time in the context of the application running. Section III describes the usage of Pig Latin to run in the modes of local and Map reduce mode. Section IV de- scribes the analysis of Map reduce execution and Pig Latin with Local and Map reduce mode of execution.

Revised Manuscript Received on May 10 ,2019

Dr K UmapavanKumar, Assoc.Professor. Dept of Computer Science & Engineering, Malla Reddy Institute of Technology, Maisammaguda,Dhulapally,Secundarabad-500100. India.

Dr S.V.N. Srinivasu, Professor, Dept. of Computer Science & Engineering, Narasaraopeta Engineering College, Narasaraopeta, Guntur, Andhra Pradesh, India.

Dr. A. Ramaswamy Reddy, Principal, **Malla Reddy Institute of Technology**, Maisammaguda,Dhulapally,Secundarabad-500100,India

II. EXPERIMENTAL SETUP

To work out the results in the paper, the following Hadoop eco system single node cluster with pseudo distributed mode have been used. Hadoop 1.0.3 stable version on top of RHEL (Red Hat Linux Server 6.0- i386) with the configuration files such as core-site.xml, hdfs-site.xml, mapred-site.xml and Hadoop-env.sh. The hdfs: //localhost: port (6789) taken as default port for Hadoop distributed file system. The localhost: port (1234) taken as port for job tracker to setup the map reduce functionality. The MR setup with the above-mentioned configuration along with 1TB HDD and 8GB RAM is used to run the unstructured data. The Pig Latin with the setup of Apache Pig version 0.11.0. (r1446324) (org.apache.pig.backend. hadoop. ex- ecutionengine.HExecutionEngine) which connects to hdfs through port 6789 and map reduce through port 1234.

III. BIG DATA SCENARIOS IN THE CONTEXT OF UNSTRUCTURED DATA

Data is everywhere, in the social media like Facebook, Twitter and LinkedIn, similarly in the context of weblogs, click streams (Amazon and Flip kart) as well as network logs[4]. The data mostly in the form of audio, video and human readable text format which is termed as unstructured data. In a high end the textual analysis of the blogs and the audio-based systems so as to respond back to the users are most commonly used unstructured context [5-8]. In the medical sector X-rays are the best example of unstructured data, to analyses the images and to diagnose the decease the unstructured formats are used [9-10].

The storage of the unstructured data is bit different when compared with structured data, and for that the Google file system is one of the best possible ways of storing unstructured data in a best possible way [11]. The processing of unstructured data is a biggest challenge and need to embed the aspects of parallel and distributed so as to speed up the data processing and reduce the availability errors [12-14]. The data considered for the research is wiki page counts data which maintains statistics of the search engine

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

data like how many members have visited a document or article during a time period. The data is huge in nature and the content which is stored is unstructured in nature and processing of the data with such background requires different storage mechanism and processing mechanism [15].

The simple example about the usage of huge amounts of the data in social media is really amazing. The sample screen shot provides the way how the data population is done in the Facebook application.

Fig 1. Facebook Data Population Source: Facebook Research

In general, the application usage in the web also strongly depends on the source of the unstructured data. Al- most all the applications based on the web strongly fetching the data in the format of log files, audio or video con- tents and human readable text. The banking sector embeds the CIBIL score to identify the genuine customers while sanctioning the loan and other benefits to the members. In case of credit card transaction to detect the fraud detection the application should work with unstructured data.

Figure 2: Unstructured data population in web applications. Source: Google

IV. MAPREDUCE CONTEXT OF DATA PROCESSING

The current scenarios of applications require huge amounts of data processing, the usage of unstructured data in various applications has been observed. MapReduce (MR) is a programming model where the data can be processed in a distributed and parallel mode of the operations. Distributed refers to reliability, replication and availability[16]. The application which involves the huge amounts of the data needs protection and reliability in the execution of the jobs. MapReduce provides distribution based on the Hadoop Distributed File System (HDFS), which is a way of storing data in reliable manner. The storage (HDFS) and processing (MR) of Hadoop framework de- pends on the daemons which are Name Node (NN), Data Node (DN), Secondary Name node (SNN), Job Tracker (JT) and Task tracker (TT) [17]. The services of HDFS are NN, DN and SNN. The services of MR are JT and TT. The other context of this framework is reference of Master/Slave architecture in the running of the daemons. The Master node instructs the slaves to perform some task; slaves follow the master and act accordingly. The daemons related to master are NN, JT and SNN, the daemons related to slave are DN and TT.

Issues Identified in the Architecture

1. The NN is a single point of failure, and SNN is not hot backup.
2. The JT is overburdened.
3. The Storage model follows 64/128/256MB fixed block size to al- locate the data.
4. The Driver code is redundant.

In the current research the solutions to above mentioned issues are not going to describe rather the scope is to com- pare and analyses the working models of the MR and Pig Latin functionalities in the implementation of the applications. The application implementation of MR conceptually involves various phases in the framework like Split Phase, Map Phase, Shuffle Phase and Reduce Phase. The (Key, Value) pair mapping is the functionality of the split phase; the business logic implementation is the functionality of the Map Phase, sorting and grouping of the data is the task of shuffle phase. The reduce phase performs the aggregation and storing of the final result into HDFS. The Map and Re- duce phases need to implement by the Hadoop Developer whereas the Split phase and shuffle phase automatically handled by the Hadoop framework. The context of programming with MR is bit distinct, the configuration of the Eclipse (Juno, Kepler.), design of Map, Reduce and Driver code, Building the path of Hadoop API which involves various .jar files of Hadoop [18].

Exporting the jar file to the file sys- tem and finally submit the Hadoop job to the terminal. hadoop jar ¡jarfile-name¿¡packagename.Driverclassname¿ HDFS I/P *HDFSO/P >*

Warning :HADOOP HOME is deprecated.
19/01/11 23:14:09 WARN mapred.

JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.

19/01/11 23:14:09 INFO util.NativeCodeLoader: Loaded the native- hadoop library

19/01/11 23:14:09 WARN snappy.

LoadSnappy: Snappy native library not loaded

19/01/11 23:14:09 INFO mapred.

FileInputFormat: Total input paths to process : 1

19/01/11 23:14:10 INFO mapred. Job- Client: Running job: job2019011122420001

19/01/1123 : 14 : 37*INFOmapred.JobClient* : *map*10019/01/1123 : 14 : 57*INFOmapred.JobClient* : *Counters* : 30
19/01/1123 : 14 : 57*INFOmapred.JobClient* :

Totaltimespentbyallmapswaiting afterreservingslots(*ms*) 0
19/01/1123 : 14 : 57*INFOmapred.JobClient* :
Data localmaptasks = 2
19/01/1123 : 14 : 57*INFOmapred.JobClient* :
$SLOTS_MILLIS_REDUCES$ = 14891
19/01/1123 : 14 : 57*INFOmapred.JobClient* :
F ileInputFormatCounters
19/01/1123 : 14 : 57*INFOmapred.JobClient* :
BytesRead = 363
19/01/1123 : 14 : 57*INFOmapred.JobClient* :
F ileOutputFormatCounters
19/01/1123 : 14 : 57*INFOmapred.JobClient* :
BytesWritten = 27
19/01/1123 : 14 : 57*INFOmapred.JobClient* :
Totalcommittedheapusage(*bytes*) = 248127488
19/01/1123 : 14 : 57*INFOmapred.JobClient* :
CPUtimespent(*ms*) = 6150

V. PIG CONTEXT OF RUNNING THE JOBS

The simple scripting can be done with the usage of Pig Latin in the eco sys- tem of Hadoop. Pig provides the environment where the scripts cab be implemented in various contexts. Pig scripts can be taken in local mode or MR mode. Pig follows a kind of architecture in the running of the scripts. Pig scripts initially submitted to the grunt shell, which in turn move the script into various logical plans so as to provide the optimization[19]. The best logical plan mapped to the physical plan and finally the physical plan is mapped to the MR plan to finalize the implementation. The script involves very less code when compared to MR implementation and development time is very less in Pig when compared to MR[20-22]. The implementation of the scripts in the pig follows the storage of the script with .pig extension and mentioning the source data either in local file system or in HDFS[23]. The output can be routed to either local file sys- tem or HDFS based on the mode of Pig running (Local, MR). Grunt pig x local wordc.pig

Hadoop Version Pig Version UserId StartedAt FinishedAt Features

1.0.3 0.11.0 hdp 2019-01-11 22:45:03

2019-01-11 Success!

Job Stats (time in seconds): JobId Alias Feature Outputs

Output(*s*) :
Successfullystoredrecordsin : "*file* : *///home/hdp/lfspig*" *JobDAtt* :
*jobIocalo*001− > *jobIocalo*002, *jobIocalo*002− > *jobIocalo*003, *jobIocalo*003
2019 01 1122 : 45 :

24, 515[*main*]*INFO* *org.apache.pig.backend.hadoop .executionengine.mapReduceLayer. MapReduceLauncher* − *Success*!

VI. ANALYSIS OF MAPRE- DUCE AND PIG LATIN EXECUTION CONTEXT

The Parameters considered in MapReduce implementation. In the execution of MR task various parameters appeared in the running of the task, such as Total input paths to process, Counters, File Input For- mat, Map Input records, Reduce shuffle bytes, Spilled Records, Map Out- put Bytes, Total committed Heap Us- age, CPU Time Spent, Physical memory and Virtual memory. The Parameters considered in Pig Latin Implementation

In the execution of Pig Latin script various parameters appeared in the running of the task, such as Start Time, Finished Time, Features such as Group By and Order By, Input path

job*Iocal*0001*grp,ecords, records, results* information,
output path information
ttROUP$_B$Y, COMBINER
*jobIocal*0002*sorted,esultsSAMPLER jobIocal*0003 *sortedresultsORDER$_B$Y*
file :///home/hdp/lfspig,
Input(*s*) : *Successfullyreadrecordsfrom* : "*/home/hdp/wcip*"
and specification of DAG.

The analysis as per the observation specifies that in MR most of the emphasis is on path of the process, in- put formats such as Text Input format, Key based input format and sequential input formats to observe the data in the format of Key, Value. The other point is consideration of Map Input and Reduce shuffle as the map phase every time emits the temporary output and the common keys must be routed to a particular reducer. The final point of preference in MR is usage of Heap memory, CPU time spent along with both physical and virtual memory usage.

In case of the Pig Latin context the emphasis is on session start time and finished time, the arrangement of data in group by and ordered by phases of the implementation. The clear mention of input and output path, the reason is to notify that Pig Latin can be implemented with either Local file system mode of operation by taking the input from the local file system LFS and putting the final outcome in LFS The same is not possible in MR.

VII. CONCLUSION

The goal of the work is to observe the parameters involved in the MR and Pig Latin implementation, successfully achieved the parameters list and described the usage of each and every parameter in the context of MR and Pig Latin. The paper mainly concludes the influence of the parameters in the context of Pig Latin and MR which helps to improve the corresponding implementations with respect to the performance in case of time and space requirements. The work also considers the background of MR conceptual tasks and technical aspects involved.

The usage of Pig context especially in LFS and MR are remarkable the same may not be achieved in the MR context. The analysis of the parameters gives the insights of Hadoop and MR along with Pig Latin script implementation. The extension of the work is to ad- dress the issues identified in the frame- work along with comparative analysis of task running in the context of Pig Latin and MR.

REFERENCES

1. Uma Pavan Kumar K, Various Is- sues in Hadoop Distributed File System, Map Reduce and Future Research Directions, International Journal of Pure and Applied Mathematics, Volume 120 No. 6 2018, 4441-4451, June 24, 2018.
2. www.dezyre.com
3. www.apache.org
4. Harish Balaji, Ujjwal Pal and Uma Pavan Kumar K., Big data Techniques and Analytics in Distributed E-commerce business, International Journal of Control theory and applications, Volume: No.9 (2016) Issue No. :3 (2016), Pages : 1719-1726
5. www.analyticsvidya.com
6. www.udemy.com.
7. www.kaggle.com
8. www.github.com
9. www.quora.com
10. www.forbes.com
11. https://machinelearningmastery.com/best-machine-learning-resources-for-getting- started/.
12. www.courseera.org
13. https://www.kdnuggets.com
14. S. Lohr, “The age of big data,” N. Y. Times, vol. 11, 2012.
15. S. Madden, “From Databases to Big Data.,” IEEE Internet Computing., vol. 16, no. 3, 2012.
16. P. Zikopoulos, C. Eaton, and others, Understanding big data: Analytics for enterprise class hadoop and streaming data. McGraw-Hill Osborne Media, 2011.
17. A. McAfee, E. Brynjolfsson, T. H. Davenport, D. J. Patil, and D. Barton, “Big data,” Manag. Revolut. Harv. Bus Rev, vol. 90, no. 10, pp. 61–67, 2012.
18. R. Appuswamy, C. Gkantsidis, D. Narayanan, O. Hodson, and A. Rowstron, “Scale-up vs Scale-out for Hadoop: Time to rethink?,” in Proceedings of the 4th annual Symposium on Cloud Computing, 2013, p. 20.
19. A. S. Tanenbaum and M. Van Steen, Distributed systems. Prentice-Hall, 2007.
20. C. P. Chen and C.-Y. Zhang, “Data intensive applications, challenges, techniques and technologies: A survey on Big Data,” Inf. Sci., vol. 275, pp. 314–347, 2014.
21. T. B. Murdoch and A. S. Detsky, “The inevitable application of big data to health care,” Jama, vol. 309, no. 13, pp. 1351– 1352, 2013.
22. I. Mashal, O. Alsaryrah, and T.-Y. Chung, “Performance evaluation of recommendation algorithms on Internet of Things services,” Phys. Stat. Mech. ItsAppl., vol. 451, pp. 646–656, 2016.
23. K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The hadoop distributed file system,” in 2010 IEEE 26th symposium on mass storage systems and technologies.

AUTHOR(S) PROFILE

Dr.Umapavankumar Kethavarapu currently working in CSE department Malla Reddy Institute of Technology, Hyderabad. Received his PhD in CSE from Pondicherry Central University,2018. His area of interest includes Big data with Hadoop, Machine Learning, Deep learning and Analytics. He has published more than 45 articles in reputed journals which includes Scopus and other indexed bodies.

Dr. S. V. N. Srinivasu, Currently working as Professor in CSE department Narasaraopeta Engineering College. Received his PhD in CSE from Acharya Nagarjuna University,2014. His area of interest includes Software Engineering, Software Testing, Machine Learning, Deep learning, Mobile Networking, Operating systems, Data Mining, Image Processing and Black chain technology. He has published more than 58 articles in reputed journals which includes Scopus and other indexed bodies. He is a member of IEEE, MIET

Dr.A.Ramaswamy Reddy Currently working as Principal in Malla Reddy Institute of Technology, received his PhD from JNTUK.. His area of interest includes applications of Image processing, Data Mining and Machine Learning techniques, Deep learning and Analytics. He has published more than 55 articles in reputed journals which includes Scopus and other indexed bodies. He is a member of many societies like IEEE, ISTE, Life Membership FIE .