Distributed Frequent Itemset Mining using Pig on Top of Hadoop Framework

Shubhangi Gawde Gurkhe
M.E. Student
Computer Engineering dept.
A.R.M.I..E.T.
Shahapur, Thane, India

Vipul Dalal
Ankit Sanghavi
professor
computer engg dept
A.R.M.I..E.T Shahapur

***Abstract*—Distributed computing environment utilizes the computing power of all the resources connected on the distributed cluster network which results in to efficient, fault tolerant, flexible working environment to process massive data. This paper focused on Use of Pig Latin language which provides ease of programming to develop Apriori algorithm on distributed cluster environment.**

Keywords—Cluster, Data node, Name node, Pig Latin, Pig Engine ,Foreach, Filter, Group, Order By, Distinct, Join.

I. INTRODUCTION

In the era of digitalization the use of internet is irresistible. Everyone is accessing internet for completing their day to day activities such as communication, online shopping, social media accessing, web searches, different software applications, gaming, internet of Things, internet as service ,sensor networks etc. As a result large amount of information from online sources, business, online transactions are generating which is very much useful if it is properly processed, analyzed and collected in terms of data sets. This motivates the development of Distributed computing system using Apache Hadoop platform.

Hadoop platform has grab the attention of almost all data processing industries as well as academics, research area. due to its features such as open source, scalability, reliable, distributed computing and fault tolerant. On the Hadoop platform usually coding is done in java programming language which is procedural language. Whereas every developer is not familiar with java programming to solve this issue Pig Latin high level programming language was invented. It serves those users who are familiar with SQL. It is a high level platform which provides execution framework for parallel computation based on Hadoop. It automatically generates Map-Reduce programs.

Rest of the paper covers sections as follows II) frequent Itemset Mining III) introduction to Pig Latin IV) system architecture , hardware & software requirement V) implementationVI) Performance analysis VII) Summary.

II. FREQUENT ITEMSET MINING

A collection of one or more items is called as itemset. eg.{milk, bread, tea powder etc.}. Frequent itemset is nothing but An itemset whose support is greater than or equal to a *minsup* threshold. According to the association rule mining an itemset is frequent, then all of its subset must also be frequent. one of the important step in the data mining is association mining which was first invented by the Agrawal[1].it involves basically two steps.

1. Find out frequent itemset which can fulfill the minimum support criteria of user specified minimum support.[1]
2. Generate the rule by applying the user define minimum confidence and frequent itemset[1].

There are so many algorithms are there to generate association rule-

- Apriori
- FP-Growth algorithm
- Eclat algorithm etc.

These are some famous algorithms for association mining.

This traditional algorithm gives the best performance when size of database is small but in case of massive data processing it needs repeated database scans which causes to detoriation in their performance. As we see the use of internet generates lots of information and it is exploding. While handling such massive data traditional algorithm are giving poor response. To handle such Big data task distributed working environment is implemented in this paper. Hadoop Map reduce framework is initially preferred but due to its coding complexity the Apache Pig Latin coded approach is suggested in this paper.

Pig Engine converts the Pig Latin script in to MapReduce statements and then mappers and reducers are invoked to execute further steps. Therefore it takes approximately 3 multiplying factor more time to execute than the MapReduce approach. to optimize the performance of the system following system is designed.

III. INTRODUCTION TO APACHE PIG

Apache .Pig Latin language allows to specify a sequence of data transformation such as merging data sets, filtering them and applying functions to record or group of records. Pig comes with many built in functions and also allows creating User Define functions to do special purpose processing. Pig Latin program runs in distributed manner on a cluster where programs is compiled in to Map/Reduce jobs and executed on Hadoop

Apache Pig has two parts are as follows:

- **Pig Latin** : A SQL like language that hides the details of MapReduce computations from programmers[3]
- **Pig Engine**: The platform that compiles optimizes and executes Pig Latin as a series of MapReduce jobs.[3]

There are 8 built in operations- Foreach, Filter, Group, Order By, Distinct, Join, Limit, and Sample. Pig also allows user define functions. Pig Latin is scripting language it doesn't support control loops or data structures

Before going to start with implementation of algorithm we should be aware of some overheads are involved in Pig script execution.

"1)Overhead of translating Pig Lati in to Java MapReduce job.2)the overhead of converting data to and from the text format used in HDFS file and Pig's own internal representation.3) Additional MapReduce job that are performed by the Pig Engine.[3]"

IV. SYSTEM ARCHITECTURE , HARDWARE & SOFTWARE REQUIREMENT

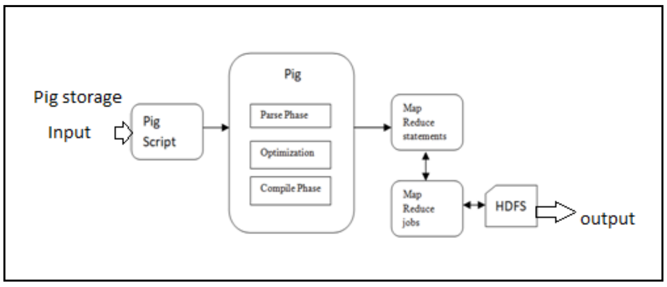


Fig 1:System Architecture

The system architecture is consisting of Master Slave architecture. The system is using Hadoop Distributed file system HDFS which works on Name nodes and Data nodes. Here in this system distributed cluster is created of two Data nodes and One Name node. Name node keeps the record of each operation done on file such as read, write, deletion etc .it also look after Data nodes health I failure found then another Data node is connected. The Data nodes actually perform the computations and replicate the result on HDFS. Secondary Name node is a checkpoint Name node it keeps all updates to the HDFS[12]. The heap size of HDFS is decided to 6MB.

Hardware	Master	i5,4GB RAM,1TB HDD
	Slave	P4,4GB RAM,500 GB HDD
	Network connection	Wi-Fi speed up to 10 MB
Software	Environment	UBUNTU 18.10
		Hadoop Pig Latin

Table 1: Hardware and software requirement

V. IMPLEMENTATION

Apriori algorithm is one of the traditional algorithm which is proposed by R.Agrawal and R.Shrikant in 1994 for mining frequent itemset for Boolean association rule. It has two deadly bottlenecks [1].

1) It needs great I/O load when frequently scans database.[1]

2) It may produce overfull candidates of frequent itemsets[1]

To resolve this problem many of the professional algorithms proposed. In this paper we are proposing Optimized Pig Latin based Distributed Apriori Algorithm.

Implementation of
Pig Latin based Distributed Apriori algorithm

1. LOAD input file in the Pig storage.
2. Apply FILTER to remove null records.
3. Use GROUP by command to extract only the desired column./columns from the database file (this step also reduce the overhead of loading full file in to the memory while processing).
4. FOREACH group find out the count for each itemset as its frequency and compare the count with *minsup* threshold.
5. FILTER the records which frequency count is less than *minsup*.
6. DUMP or STORE the list of frequent Itemset.
7. Next take the (key, value) pair of previous step output and apply Replicated join on the database itself.
8. Apply FILTER to remove the null present in join records.
9. Now apply the FILTER to find out frequently occurring pair of itemset which fulfill *minsup* threshold condition.
10. Now take Replicated join to the output obtained at 4th step and output obtained at 6th step and obtained three itemsets along with their frequency.
11. This will be repeated till frequent itemset is empty.

In above "Optimized Pig Latin based Distributed Apriori algorithm" LOAD, GROUP, FOREACH, FILTER, Replicated are all keywords from Apache Pig library used for scripting .

As explained in the end of section III of this paper Pig performance need to be improved so following section explains our contribution in optimizing the Pig based frequent item set mining algorithms performance.

Firstly In step number 2 we have applied null key to the records loaded in to Pig storage. Here it drops the all null records that may exist in the input file.

- The ' null' key removes the unnecessary burden of processing the null records which speed ups the filtering step ahead. Hence the execution time reduces.
- As well as before join we are removing null records which will remove the burden of extra mapping and reducing steps at the time of join. Hence it will also improve the performance.

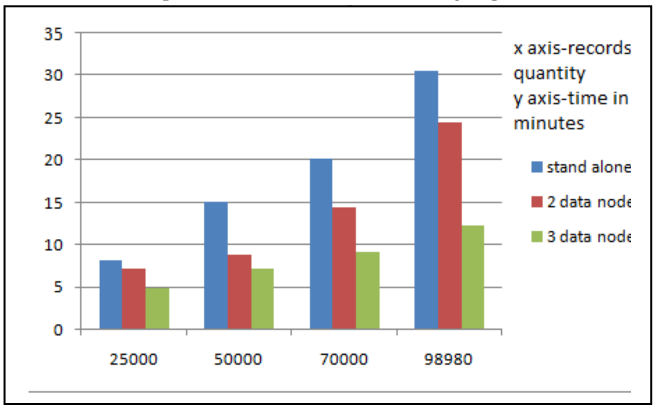
Secondly we are applying 'Fragment replicated join' which is a special type of join. By using this all the hadoop work is done on the map side so it helps to reduce the total number of

MapReduce jobs and hence it improves the overall efficiency of the algorithm.
We have established a cluster set up of two data nodes and one name node and observed the following results.

VI. PERFORMANCE ANALYSIS

No. of records	stand alone	2 data node	3 data node
25000	8.2	7.2	5
50000	15.2	8.78	7.18
70000	20.23	14.5	9.24
98980	30.6	24.5	12.15

Table 2: Required execution time as varying number of



Above graph shows the performance of Pig based distributed Apriori algorithm. Table 2 shows the results we have obtained in the distributed mode and standalone mode.as we know the Pig engine requires to do extra overhead of converting Pig Latin script in to MapReduce format. Therefore in standalone mode of execution as number of records increases the execution time also increases. to overcome this issue we have implemented distributed mode .In distributed mode we have observed that the output with 2 data nodes the execution time reduces as shown in the table 2.simillarly if we add one more node that is total 3 data nodes then execution time again reduces .In this manner distributed mode helps to reduce the execution time. As many number of data nodes are added the execution time will reduce. However the performance also depends on the number of records or how big the input file size, the hardware configuration of the nodes and network speed .

VII. SUMMARY

We have tested above algorithm on the distributed cluster nodes .here we have observed that as we go on adding nodes in the cluster the execution speed gets improved. While in standalone mode takes longer time for the execution and some times hangs also. Unlike the hand coded MapReduce code Pig Latin script takes more time for the execution due to the overheads present as we have discussed in the section number III of this paper.
Distributed mode of execution is more cost effective and reliable and fault tolerant .if one or more nodes in the cluster goes down then Name node immediately manage the data traffic connected to them and allocate another nodes.
Fragment replicated join performs well and improves the efficiency of the algorithm but the limitation is the file size should be small enough to in memory.

REFERENCES

[1] R.Agrawal, and A. Swami-, “Mining association rules between sets of items in large databases”. *In: Proc. of the l993ACM on Management of Data, Washington, D.C, 207-216, May 1993.*
[2] https://pig.apache.org/docs/r0.17.0/perf.html#Use+OptimizationK. Elissa, “Title of paper if known,” unpublished.
[3] https://pdfs.semanticscholar.org/ec19/1d78641c81fcc7e744cded3a0e88 47fe86a4.pdfress.
[4] Z. Chen and S. Cai “ An Improved Apriori Algorithm Based on Pruning Optimization and Transaction Reduction”, *in IEEE International Conference ,978-1-4577-0536-6/11, Dec 2011 .*
[5] JDean and S.Ghemawa”MapReduce:Simplified data processing on clustes”.pp 137-150 OSDL 2004.
[6] C Olston, G Chiou, L Chitnis, F Liu, Y Han… - Proceedings of the …, 2011 - researchgate.net
[7] 2017 International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT)
[8] 2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010)
[9] https://www.researchgate.net/.../281403776_Big_Data_And_Hadoop_A_Review_Pap .
[10] L Li, M Zhang - … conference on business computing and global …, 2011 - ieeexplore.ieee.org
[11] MY Lin, PY Lee, SC Hsueh - … of the 6th international conference on …, 2012 - dl.acm.org
[12] https://hadoop.apache.org/