International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 01, January 2022)

An Optimization Method for Performing Operation on Map Reduce

Manoj Saini[1], Upendra Varshney[2]

Department of Master of Computer Application, Compucom Institute of Information Technology and Mangement

***Abstract*—There is a significant number of data that needs to be stored and processed nowadays as internet technology develops quickly. This developing technology generates data, of which only 20% is structured and 80% is unstructured, which is known as the "big data dilemma."**

Therefore, there is a demand for technologies that can effectively manage massive data. Large data sets can be processed in a distributed fashion using the Apache Hadoop platform. Having two modules in TheHadoop Map Reduce and the Hadoop distributed file system. The data will be handled more quickly with Hadoop and an increasing number of nodes when utilising the hive query language.

This article explores the MapReduce Join operation, which is used to merge two huge datasets. The process of joining two datasets begins by comparing their sizes. In this paper, we proposed a distributed cache, so the smaller dataset is stored into the cache memory of each mapper node and all the lookups are performed by mapper to produce a final records. If one dataset is smaller than the other dataset, then either Mapper uses the smaller dataset to perform a lookup by Reducer for matching records from the large dataset and then combines ethoser records to form output records. Therefore, by utilising distributed cache, data load will be managed well at mapper ends, resulting in overall superior performance compared to regular join.

***Keywords*—Bigdata, Hadoop, Mapreduce, performance, Hive, Query Optimization,**

I. Introduction

Big Data is the term used to describe the collection of data sets that are so complex and massive that they are challenging to analyse using conventional management or processing techniques. Volume, Velocity, and Variety are the three V's that Gartner uses to describe big data in terms of the resources needed to handle it [1]. Additionally, other Vs including Validity, Veracity, Visibility, and Value have been introduced by certain academics to clarify Big Data

Data is processed and analysed using a variety of applications and tools, including Hadoop, an open-source Map-Reduce project funded by Yahoo that first appeared in 2006 and is used to process Exabyte or Zettabyte-sized amounts of data on a cluster of commodity hardware connected by ethernet cables [3][4].

Hive may be an Apache Foundation data warehousing tool built on top of the Hadoop distributed framework that will handle and query data stored in HDFS in a way similar to that of traditional management systems (RDBMS). Originally created by Facebook, Hive is currently used and developed by others business models like Netflix [5]. Hive also leverages HDFS for storage and provides convenient data retrieval for users as if they were using a typical database engine, as the Hadoop distributed file solution has been the easiest option for parallel, execution, and aggregating flatfiles [6]. Process unstructured data as if it were structured using Hive Derby Language (No-RDBMS schema). Hive processes data and stores it in various softables and partitions that can be accessed using a command language exclusive to Hive called HiveQL, which is quite similar to SQL and is undoubtedly handled by those familiar with traditional management systems. Hive being comparatively small Hive is still not in a stable condition, therefore query optimization may be a topic that comes into emphasis for young projects. As a result, developers are improving the performance of Hive at this development stage. Performance of any database engine are often measured by its reaction time and amount of labour done by it. At an equivalent time, this language also allows programmers who are familiar with the MapReduce framework to plug in their own mappers and reducers programmes to perform more sophisticated analytical tasks.

II. Literature Review

Hadoop may be a widely used open-source map-reduce implementation for storing and processing extremely big datasets, but using Hadoop is difficult for end users, especially for those who are unfamiliar with the map-reduce methodology. Users must build map-reduce algorithms even for simple tasks like obtaining raw numbers or averages. Users may easily query, summarise, and analyse Big Data with SQL-like expressions called HiveQL thanks to Apache Hive, which supports a number of file formats for data input and export to and from the storage filing system. Hive's goal is to make processing petabytes of data simple and effective. Unlike RDBMS, Hive stores data throughout time.

Document-based structures that use joins have poor performance and use a lot of resources. However, by correctly setting Hive, it is possible to improve performance for the relational data. In this study, we employ a number of optimization strategies to improve query performance and assess the outcomes to reflect their impact on the outcome.

According to [2], distributed Big Data storage, specifically Hadoop, has attracted more attention than ever due to the development of ever-expanding online services, and as a result, fundamental criteria like fault tolerance and data availability have become a worry for these platforms. In Big Data applications, data replication strategies are moving toward dynamic techniques based on the recognition of files. The dynamic replication factor's formulation made it possible to address the issues it had caused

But from the empirical facts, it can be inferred that file popularity is temporary rather than permanent in nature and, after a given period, content's popularity ceases most of the time, introducing the I/O bottleneck of updating replication inside the disc.

In [3], indexing offers quick searching over large amounts of data. It is a data structure strategy to effectively retrieve entries from database files supported by some indexing-related features. With the assistance of indexing, queries typically lead to much better results.

Performance oriented data ware houses software. We will carry out the work of query processing and data analysis with the aid of hive. Hive is well-liked because it supports the majority of SQL procedures in electronic database administration systems. Many query optimization strategies have focused on improving database system join performance. As a result, this study has two goals: first, we develop the index-based join technique and integrate it into Hive, and second, performance is predicted after performing the join operation.

III. PROBLEMDEFINITION

To store, retrieve, and analyse big data effectively, a relational database is used. The relational database used is the Hadoop architecture, which includes Hive, HDFS, and MapReduce. The queries must be executed, and the execution times are recorded, after the data has been placed into the HDFS storage space offered by Hadoop. A database schema must also be constructed, and the data must be loaded into the database utilised by Hive, which offers an interface for the SQL-like syntax.

On large-scale data clusters, MapReduce is a well-liked programming paradigm for carrying out time-consuming analytical queries as a variety of tasks.

Many chances for sharing scan and/or joincomputation tasks might exist in contexts where numerous queries with comparable selection predicates, commutables, and join jobs arrive simultaneously. As a result, overall performance suffers.

IV. CONCLUSION

In this paper, we proposed a distributed cache where the smaller dataset is stored into cache memory of each mapper node and allthe lookups are performed by mapper to produce a finalrecords. By using distributed cache we didn't use reducer means data load would be managed.

REFERENCES

[1] MelihGunay; M. Numan Ince; Alper Cetinkaya"“ApacheHivePerformanceImprovementTechniquesforRelational Data” in IEEE 2019.

[2] Pinchao Liu, Adnan Maruf, Farzana Beente Yusuf,LabibaJahan,HailuXu,BoyuanGuan,LitingHu,Sitharama S. Iyengar “Towards Adaptive Replication forHot/Cold Blocks in HDFS using MemCached” in IEEE2019.

[3] Akshay Kumar Suman, Dr.ManasiGyanchandani“ImprovedPerformanceofHiveusingIndex-BasedOperationon BigData” in IEEE 2018

[4] http://www.hadooptpoint.com/introduction-hive/

[5] Yue Liu1,6,7 , Songlin Hu1 “DGFIndex for SmartGrid:EnhancingHivewithaCost-EffectivemultidimensionalRangeIndex”40thInternationalConference on Very Large Data Bases, September 1st -5th2014, Hangzhou, China.

[6] ANTLR[Online].Available:http://www.antlr.org/

[7] An, M., Wang, W., Wang, Y., “Using Index in theMapReduce Framework, ”, 12th Intl. Asia Pacific WebConf.(APWEB),Beijing, China,2010, pp.52-58

[8] Dean,J.,Ghemawat,S.“MapReduce:SimplifiedDataProcessingonLargeClusters,”Mag.Commun.ACM50thanniversary,vol.51,issue1,2008,pp.107-113

[9] Capriolo,E.,Rutherglen,J.,Wampler,D.ProgrammingHive:DataWarehouseandQueryLanguagefor Hadoop,1sted, O'ReillyMedia,2012

[10] TPC-H[Online].http://www.tpc.org/tpch/

[11] HIVE 1694[Online]. Available:https://issues.apache.org/jira/browse/HIVE-1694

[12] Hiveindexdesigndoc[Online].Available:https://cwiki.apache.org/confluence/display/Hive/IndexDe

[13] HiveJIRA[Online].Available:https://issues.apache.org/jira/browse/HIVE

[14] HIVE-1644 [Online]. Available:https://issues.apache.org/jira/browse/HIVE-1644

[15] N.Jain,L.Tang,“JoinstrategiesinHive”,Facebook,Rep. Hadoopsummit2011,2011 [Online].

[16] Li,Z.,Ross,K. A.“Fastjoinsusingjoinindices”,inTheInternationalJournalonVeryLargeDataBases,vol.8, issue 1,1999, pp.1–24