Comparing Implementation Features of Map Reduce in RDBMS with Distributed Cluster

Mohammed Muddasir N,
Assistant Professor
Dept of IS&E
Vidyavardhaka College of Engineering, Mysore

Ranjitha H C
Student
Dept of IS&E
VVCE,Mysore

Meghana S
Student
Dept of IS&E
VVCE,Mysore

ABSTRACT
Data processing techniques are becoming more innovative as the amount of data grows. Here we are exploring such techniques to process big data one is the traditional RDBMS approach and the other distributed approach. We came across certain advantages and disadvantages of both the approaches. RDBMS is a very highly used technology for data processing by various organizations and replacing it with new technology has a lot of challenges. Distributed processing is the need of the hour and technologies like Hadoop, map reduce etc. [1] is being used for processing Big Data. There is a debate on which technology to use for processing data and we have just explored some possible results measuring both the technologies.

General Terms
Big data, map reduces, Hadoop, distributed processing.

Keywords
Distributed processing, RDBMS, Hadoop, map reduce.

1. INTRODUCTION
We live in data age. In real world we come across many scenarios, where Data creation is occurring at a record rate. In 2010, the world generated over 1ZB of data; by 2014, we will generate 7ZB a year. Much of this data explosion is the result of a dramatic increase in devices located at the periphery of the network including embedded sensors, smartphones, and tablet computers. All of this data creates new opportunities to "extract more value" in human genomics, healthcare, oil and gas, search, surveillance, finance, and many other areas. We are entering the age of "Big Data". Big Data is defined by three V's volume, velocity and variety [2]. Volume is the huge amount of data which is required to be stored and processed. Velocity is the speed at which the data is generated according to statistics only some percentage of entire world population is on the internet generating data by years to come almost 50% of the world population will be in the internet generated huge amounts of data at a very high speed. Variety refers to the different sources such as internet applications, smart phones, social media and forms of data such as text, image, multimedia etc. Today 4th V referring to veracity is used to define quality of big data.

Often data scientist and analyst are happy that we have big data but they say we do not have a solution for analyzing this data to extract useful information from this data. Going in the direction of finding a solution to store and analyze big data first we have Google's GFS Google distributed file system which was an open paper presented in 2004. Tom White at yahoo implemented the distributed file system called it HDFS which is Hadoop distributed file system [3]. There are many other solutions such as green plum from EMC2 and BigInfoshpere by IBM. Tom White writes in his book "Hadoop a definitive guide" [1] we have storage technologies improved but still have to work on transfer speed. In the 90's storage capacity was 1370 MB and transfer speed was 4.4 MB/s so we could read all the data in 5 minutes. Now the storage is in the capacity of terabytes and zeta bytes and transfer rate is 100 MB/s so it takes nearly 2 to 3 hours to read all the data in the disk. The solutions are focused on breaking the data into blocks and storing blocks into various hard drives in a distributed environment. Later when we require reading the data make all the processors in the distributed environment work parallel to exact the blocks so we could read a 1TB of data in say about 2 to 3 minutes instead of 2 to 3 hours.

Exploring the various solutions for big data we came across the paper "oracle in database Hadoop" [4] which is using the traditional RDBMS for processing big data. The common thing among "Oracle in database Hadoop" and HDFS is they both are using a programming paradigm called Map Reduce. The difference is in the paper "oracle in database Hadoop "they claim a distributed environment is not necessary for processing small amount of data. We had a thought any ways small amounts of data are processed efficiently by existing RDBMS including oracle. If we are not using distributed processing then we are limited by the transfer rate of the disk with whatever programming paradigm we use. The point is if we are looking for processing huge amounts of data we need to have a distributed processing. Hence we took up this project to compare the capabilities of "oracle in database Hadoop" and HDFS.

We compared the results of processing structured and unstructured data in HDFS and "oracle in database Hadoop" and we came to know "oracle in database is Hadoop" is faster than HDFS. But we have many results that are necessary for observation before using "oracle in database Hadoop".

2. LITERATURE REVIEW
Data storage started with traditional file based systems. Then came the RDBMS which is a software helping to store data. These systems are depending on the processing capability of single computer or processing unit. Distributed processing was the need of the hour. RDBMS vendors came up with distributed solution but those were expensive. HDFS that is Hadoop distributed file system came as a silver lining for large scale data processing. It does not require expensive hardware and could be run on commodity hardware. It is fault tolerant and scalable. Because HDFS architecture uses commodity hardware these hardware can fail hence the design of HDFS is such that it will with stand the hardware failure. This is done using replication. Also HDFS uses processing capability of the systems connected in a distributed environment and hence reduce the latency time. These systems are scalable in the sense we can add on any number of systems and more we add the computer it would have greater processing powers. HDFS main focus is to reduce the latency and process huge amount of data

faster. One more addition to the suite of distributed processing is a programming language that suits the requirement of processing huge amount of data which is Map Reduce. Map Reduce is a paradigm for processing huge data in parallel on a distributed environment. It can be used for processing both structured and unstructured data. Map Reduce has two functions called as Mapper and Reducer these work with key value pairs. If we take a simple example of counting number of words in a file Mapper job is to identify a word in the document. Reducer job is to find similar words in the document and increment the counter indicating the number of times this word has accorded in the document. Map Reduce on a distributed environment works on the philosophy of sending the processing module to the data as compared to the previous concept of moving the data to the processing unit such a program or a process. Map Reduce has been proved extensively with Hadoop in the open distribution of Hadoop. Map reduce has been implemented in JAVA for traditional java developers. It could be written in HIVE a query based language for developers familiar with query based language. It could also be implementing in script based language called PIG. In this paper we are using PIG for running map reduce job. In case of oracle we are using the SQL language for writing scripts that run map reduce job on an oracle database.

3. RELATED WORK

As it is a tough choice whether to go with existing RDBMS or implement Hadoop or a solution which give the best of both the technologies we came across various studies. Most of these studies are comparing Hadoop with parallel DBMS.

One such study "A comparison of Approaches to Large Scale data Analysis" [5] compares Hadoop with parallel RDBMS. It claims Map reduce to exist in parallel SQL. The study compares 2 DBMS with MR. In this study to run MR programs they have used Hadoop which even have HDFS for storing data. They use a 100 node cluster and compares performance and development complexity. The observation is it takes more time in parallel RDBMS to load the data and tune as compared to MR. But time to execute the query on a 100 node cluster is faster in parallel DBMS then in MR. Although it's not appropriate to compare performance of parallel DBMS with MR on 100 node cluster because the actual use of MR is on clusters that are at least 1000 nodes or more to store petabytes of data. The author's gives example of eBay using Teradata configuration using 72 nodes to process 2.4 PB of data with 300GB of data storage on each node and 32GB RAM. The authors claims that 1000 nodes clustered are not required in practice because very few datasets actually cross PB of data. They take various parameters such as schema support, indexing, programming model, data distribution, execution strategy, flexibility and fault tolerance to compare the systems. They argue on various advantages and disadvantages of the above parameters with DBMS and Hadoop MR.

Another study "Analyzing Massive Astrophysical Datasets: Can Pig/Hadoop or a Relational DBMS Help?" [6]Takes actual scientific data and compares which system could be better. They used DBMS and PIG/HADOOP system to store and manipulate data got from simulation of large scale formation and evolution of structures in the universe. Total amount of data generated by each simulation was 55GB to some TB. They performed filtering, correlating and clustering data. The study used both technologies DBMS and Hadoop to compare the performance of each for analysis of the data. They came up with some pros and cons in both the systems. They used 2 dimensional table structures to store the data, created 3 tables to store data for one species of particles. They run basic SQL queries such as returning of particles whose property is above a certain threshold. Later they used parallel RDBMA to store data in a cluster for distributed processing and compared the results. They further used pig/Hadoop to store and run the query. They used 128MB RAM to store data and data sets were accessed frequently. Advantage noted was to store data crossing 128MB both parallel DBMS and pig/Hadoop were good. Finally concluded that parallel DBMS is faster in processing for small amount of data and Hadoop is scalable well when the data grows.

In one more study "A performance Comparison of Parallel DBMS and Map Reduce on Large Scale Text Analytics" [7] they have compare the performance of information extraction that extracts structured data from text in MR and parallel RDBMS. This work focuses on response time of information extraction as compared to other studies that focus on query processing. They build a bench mark for comparing parallel RBBMS and MR. This benchmark includes statistical learning based and rule based information extraction. They used 3 types of IE extensively used in many real world information extraction tasks. They are learning-based extractors and conditional random fields (CRF's), regular expression based extractors and dictionary matching based extractors. Since processing in parallel RDBMS required structured data they tokenized articles, used sentence splitter and finally used part of speech tagger to tag each token in the sentence. They created two tables, sentences and tokens. They used the copy command for vertica to copy data from file to DBMS and copyfromlocal to Hadoop command to copy from local to HDFS. Their results show loading times in vertical DBMS is very high as compared to Hadoop. Hadoop is much faster in loading. They have used a 16 node cluster. Information extraction performance of vertica is much superior toHadoop for a cluster of 16 nodes. This information is also proved in the previous papers we mentioned here i.e. for 100 node cluster structured data.

In our paper we are performing both structured data analysis and unstructured data analysis using pig/Hadoop and oracle in database Hadoop which is a flavor of oracle 12c having packages to run SQL query as a Map Reduce[8] query using some java functionality. We implemented both Oracle in-database Hadoop and Hadoop cluster. There we encountered certain Comparative advantages and disadvantages of both the infrastructures.

4. IMPLEMENTATION

Oracle In-database Hadoop:

The software implementation of Oracle In-database Hadoop spans across both Java and SQL domains. To run a simple word count program, we cannot use map-reduce query alone. It requires the mapreduce_pkg package that defines the input data types from the SQL side, which are used in the definition of the pipelined table functions. The package in the Java domain is the core of the framework. So we wrote map reduce-header package and map reduce-body package to support map reduce query. This makes programming much complex. As there are many rounds of context switching between SQL and Java environments, the framework needs to be able to pass the configuration parameters back and forth between theSQL and Java domains. That takes more time in executing the queries. Even in the data loading and data tuning, it consumes more time. This makes it less efficient.

Hadoop cluster

To set up the distributed environment we have taken five systems, out of which one is master node and other four are slave nodes. To implement Hadoop features we have followed certain steps. The first one was password less login, so that we can access the system in cluster without password which reduces the time. To access the system in cluster each time, we need to enter the password. To avoid that we need to set password less login from master node to slave node and vice versa using Public Key SSH Authentication.

Then we installed and configured Hadoop in five systems. So that we can store data in Hadoop distributed file system [9] and can use map reduce as programming model. We have used Hadoop 1.2.1. After the configuration

In master node, we have name node, secondary name node and job tracker

1. Name node: The name node in Hadoop is the node where Hadoop stores all the location information of the files in HDFS that is it stores the metadata whenever a file is placed in the cluster a corresponding entry of its location is maintained by the name node.
2. Secondary name node: The secondary name node is responsible for periodic housekeeping function for the name node. It only creates checkpoints of the file system present in the name node.
3. Job tracker: The job tracker is responsible for taking in requests from a client and assigning task tracker with tasks to be performed.

In slave node, we have data node and task tracker

1. Data node: Data node is responsible for storing the files in HDFS. It manages file blocks within the node. It sends information to the name node about the files and blocks stored in that node and response to the name node for all file system operations.
2. Task tracker: It's a daemon that accepts the tasks (map, reduce, shuffle) from the job tracker. It starts and monitors the Map & Reduce Tasks and sends progress/status information back to the Job Tracker.

In the browser we can check for live nodes and dead nodes. For every 3 seconds all data nodes will send heartbeat message to inform that they are alive to the master node. After creating the directories and files in HDFS, we can browse the file system for these directories and files.

Runthe Pig scripts in both local mode and Hadoop mode. To interact with the data that are stored in HDFS we are using Pig scripts.

We have installed pig-0.11.1. After successful installation we will get grunt shell where we can execute the pig scripts to get the knowledge out of data. In comparison we don't need any map reduce packages for programming since we are using Pig Latin scripts [10] [11] [12]. It provides built-in operators with

Which users can encode complex tasks comprised of multiple interrelated data transformations as data flow sequences, making them easy to write and maintain. So we found it is easy to write pig scripts for complex problems also. Here we also observed that it takes less time when we load the data to HDFS. Also by using Pig scripts programming time was reduced.

5. RESULT

We have identified several advantages of Hadoop cluster when comparing it with oracle-in-database Hadoop as shown in the table 1.

Table 1

ORACLE-IN DATABASE	HADOOP CLUSTER
Processes only structured data	Processes both structured and unstructured data
Insitu Processing	Distributed Processing
No tolerance for software and Hardware failure	Tolerance for software and Hardware failure
Only map reduce is involved	Both HDFS and Map reduce is involved
Map reduce Query is difficult and time consuming	Pig scripts are easy to understand and consume less time to implement
For large data, it's not time efficient	For large data it is time efficient
Data analysis or querying is done in small scale	Data analysis or querying is done in large scale
Oracle12c can be plugged into cloud which makes it more Expensive	Commodity hardware usage makes extreme low cost per byte whilereading and writing.

6. CONCLUSION

Big data is the data that exceeds the processing capacity of conventional database systems. The data is too big, moves, too fast or does not fit the structures of the database architecture. To gain the value from this data we must choose an alternative way of processing it, which can be done by various infrastructures like oracle-in-database Hadoop, Hadoop cluster, IBM BigInfosphere, clouds, etc. In this paper we have found many advantages and disadvantages of both Hadoop cluster and Oracle In-database Hadoop. It is at the discretion of the individual or company to use the technology based on the need for processing data. Finally we conclude that if it's processing for medium sized data traditional RDBMS could be sufficient if its huge size data then any programming paradigm will not be able to improve on the latency and we require a distributed processing infrastructure. Distributed infrastructure comes with security risk. Providing security to data on a distributed environment is a challenge. More over the data is replicated and further toughens the task. Encryption algorithms are used but they slow down the process of storing and retrieving.

7. ACKNOWLEDGEMENT

We thank the management of VVCE for providing all the support in carrying out this work. We thank our beloved principal who always encourages in research activities. Also we thank of HoD IS&E for his support and motivation.

8. REFERENCES

[1] Tom White |Hadoop: The Definitive Guide

[2] A community white paper developed by leading researchers across the United States | Challenges and Opportunities with Big Data

[3] DhrubaBorthakur |The Hadoop Distributed File System: Architecture and Design

[4] Xueyuan Su | Garret Swart |Oracle In-Database Hadoop: When Map Reduce Meets RDBMS

[5] Andrew Pavlo |Erik Paulson |Alexander Rasin |Daniel J. Abadi |David J. DeWitt |Samuel Madden |Michael Stonebraker |A Comparison of Approaches to Large-Scale Data Analysis

[6] Sarah Loebman |Dylan Nunley |YongChul Kwon |Bill Howe |Magdalena Balazinska |Jeffrey P. Gardner |Analyzing Massive Astrophysical Datasets: Can Pig/Hadoop or a Relational DBMS Help?

[7] Fei Chen |Meichun Hsu |A Performance Comparison of Parallel DBMSs and Map Reduce on Large-Scale Text Analytics

[8] Jeffrey Dean Sanjay Ghemawat |Map Reduce: Simplified Data Processing on Large Clusters

[9] http://hadoop.apache.org/docs/r0.18.0/hdfs_design.pdf

[10]http://lintool.github.io/MapReduceAlgorithms/MapReduce-book-final.pdf

[11] Dr. Hiren D. Joshi |Tapan P. Gondaliya |Big Data challenges and Hadoop as one of the solution of big data with its Modules

[12] http://www.mananing.com/lam/SampleCh10.pdf