International Journal of Interdisciplinary Innovative Research &Development (IJIIRD)
ISSN: 2456-236X
Vol. **05** Issue **01** | **2020**

Map Reduce: Data Processing on large clusters, Applications and Implementations

Bharath R.
BCA Scholar, Jain University, Karnataka, India

ABSTRACT

In the past few years, doing the search of massive amounts of data finds "hidden"and valuable information within it growing.his can result in heavy processing in considerable data,leading to the development of solutions to process such huge information based on distributed and parallel processing.MapReduce is a programming model and associated implementation for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, reducing the function that merges all intermediate values associated with the same intermediate key. The Mapreduce framework has been caught by many different areas. It is presently a practical model for data-intensive applications due to its simple interface of programming,high scalability, and ability to withstand the subjection to flaws.Also , it is capable of processing a high proportion of data in distributed computing environments(DCE).MapReduce, on numerous occasions,has proved to be applicable to a wide-range of domains.

Implementation of a mapReduce runs on a large cluster of highly-scalable machines. A typical mapreduce machine starts from lower highly scalable data like terabytes of data on thousands of machines.programmers find it easy to use ,writing hundreds of programs are implemented and upwards of one thousand mapreduce jobs are executed by google clusters.

Keywords: ***MapReduce implementations (Hadoop, Hadoop+, Grid gain, Mars, Phoenix), Mapreduce Applications (Distributed Grep, Word Count, spark, Mobile Sensor Data)***

1. INTRODUCTION

Data the information of a object or person.The asset of an it industry,which specifies the object Nowadays, the data is growth of information and data has become a burdensome challenge. The collection of data in real-world has become excessive, the collection of cluster data by the user and arrangement of the data becomes higher and higher every day.MapReduce is a fault-tolerance,simple,and scalable framework for data processing that enables its users to collect massive amounts of data.MapReduce a framework which was released by google in 2004 to tackle large amounts of data with reference to internet-based applications. Map reduce a algorithm were we record the input datasets,where "map" will receive key/value pairs and processed into intermediate key/value pairs, and then where the "reduce" will combine the all same key ,combine the pairs appropriately.Mapreduce is a simple and powerful interface that enables automatic parallelization and distribution of large-scale computations,combined with an implementation of the interface that archives high performance on large clusters. Use of functional models with user-specified map and reduce operations allows us to parallelize large computations easily and re-execute as the primary mechanism for fault-tolerance.

The easy availability accessibility of the mapreduce platforms such as hadoop,makes it sufficient for a productive parrelization and execution of data -intensive tasks.Map reduce can automatically run applications,parallel cluster of hardware and can process petabytes of data rapidly and effectively.the large input data need to be assortment,cache,fetch,inspect and also mines to allow a simple and continuous access to these data and information. MapReduce is a open-source implementation hadoop,that has become highly popular.hadoop is a open-source software utilities delivers a distributed implementation of data storage which can be scalable massive amount of data

2.PROGRAMMING MODEL

Mapreduce computation takes a set of input/value pairs and produces a set of key/value pairs. The mapreduce library expresses the computation as three functions: Map, reduce. The map function inputs pairs and produces the intermediate key/value pairs the map reduces library groups altogether intermediate pairs and assigns to the reduce function for further process. The Reduce function, also written by the user, accepts the intermediate key/value pairs and a set values for that key.it merges together all the intermediate same keys to form a smaller set of values. Typically, just zero or one output value is produced per reduce invocation. The intermediate values are supplied to the user's reduce function via an iterator. This can fit to large amount of data to fit to memory.

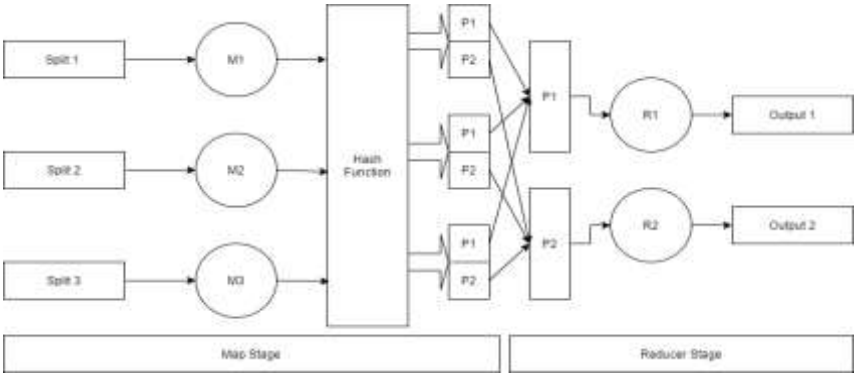

Fig 1: Mapreduce process

3. MAPREDUCE IMPLEMENTATIONS

A well-known framework in hadoop.In the IT sector mapreduce has made a tool for cloud computing applications like Yahoo,Facebook,Amazon,and IBM as implemented hadoop which is produced by Apache software foundation.Some implementations like Dryad,phoenix,Mars,Twister,GridGain.this section will provide the overview of existing implementations of Mapreduce.

3.1 Hadoop

Hadoop is a open-source software for reliable,scalable,distributed computing.Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using single programming models.It is designed to scale up from single servers to thousands of machines,each offering local computation and storage.Rather than rely on hardware to deliver high-availability,the library itself is designed to detect and handle failures at the application layer,so delivering a highly-available service on top of a cluster of computers,each of which may to be prone to failures.MapReduce of other implementations are Hadoop+,Mars,Gridgain,phoenix. It is applicable for performing the cloud-based large-scale data-parallel applications by providing the reliability and data transfer capabilities.

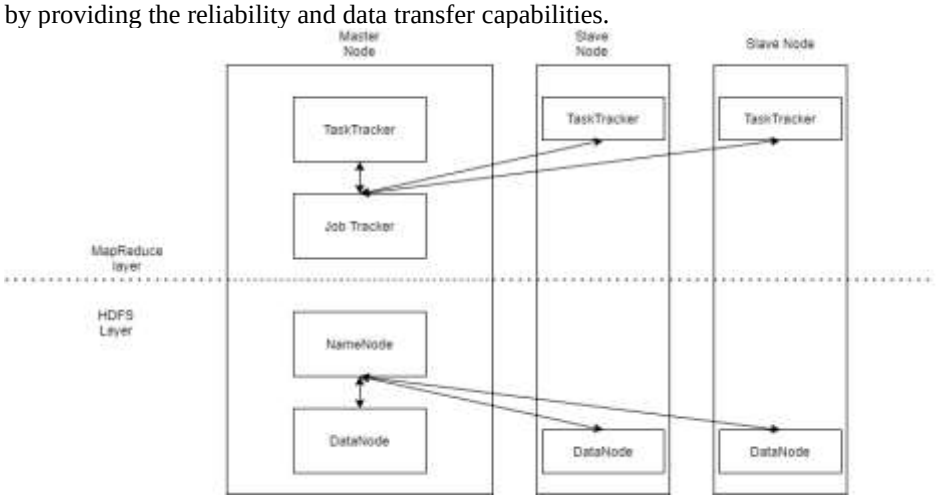

Fig 2: Mapreduce Hadoop process

3.2 Hadoop+

Hadoop is a heterogeneous MapReduce Framework which enables GPU and cpu for processing big data and leveraging the heterogeneity model to assist users for computing different resources for different purposes.Hadoop provides the map and reduce framework and primitives of Pmap and Preduce framework by Hadoop+. The pmap and prejudice functions can allow to write explicit parallel CUBA/OpenCL functions by GPUs plug-ins. In Hadoop we use map and reduce functions, but in Hadoop+ we can use Pmap , map or both, Preduce,reduce or both. In Hadoop+ can provide different input parameters in map and Pmap the explicit parallel functions.The input of map will be (key.Value)Pairs,while in PMap the input will be dataset(Collection of (key,value).Reduce and Preduce are the same parameters, the outputs are same as the Map tasks.

The Pmap family consist of parallized versions of the common list.the default parallel -mapped parts is the number of worker threads.All the Pmaps accept :parts keyword argument for specifying the number of parts explicitly. Preduce is a parallel version of reduce.It hash the input sequence into N parts and, in parallel,calls reduce on each part.the N parallel parts are then reduced again.

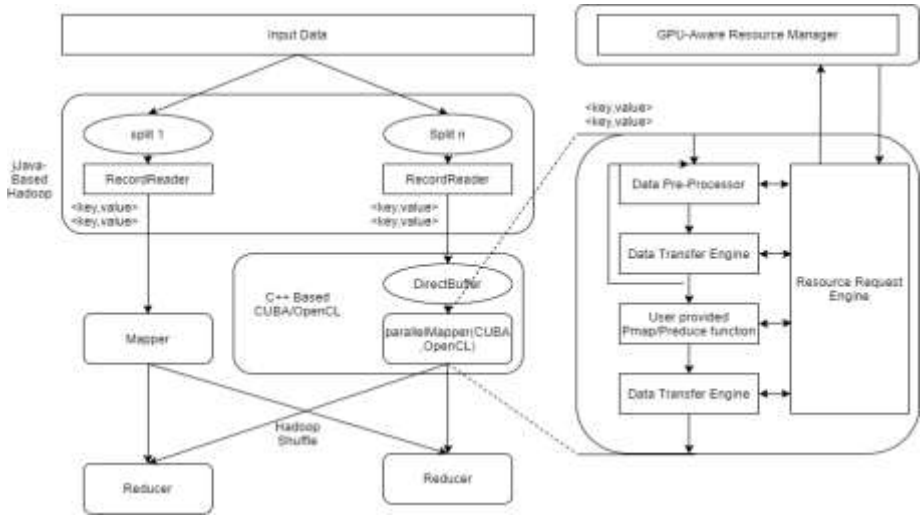

Fig 3: Mapreduce Hadoop+ process

3.3 GridGain

Gridgain a n open-source grid computing in java.A mapreduce implementation. An in-memory platform built in Apache Ignite that provides speed and massive scalability of data between applications to access across datastore and applications.The gridGain and hadoop DFS are compatible and it provides the substitute of mapreduce. When we talk from the technical point of view, in grid gain the map tasks are assigned as nodes. It acts as a bridge based on java in processing of the data and high performance of data and finishes the mapreduce implementation. In map reduce algorithm the tasks are split into the subtasks and workers will drag the split to process the data,where the gridgain will male the subtasks into nodes and process the data faster and the delivery of the tasks become easier and the time of the process will be saved. In gridgain the programmers will process the gigabyte of the data in a few seconds.this proves to be an load-balancing capabilities. This can benefit according to the situation of the developer.it can introduce complexity that developer has to plan advance so that the worker does not stay idle.

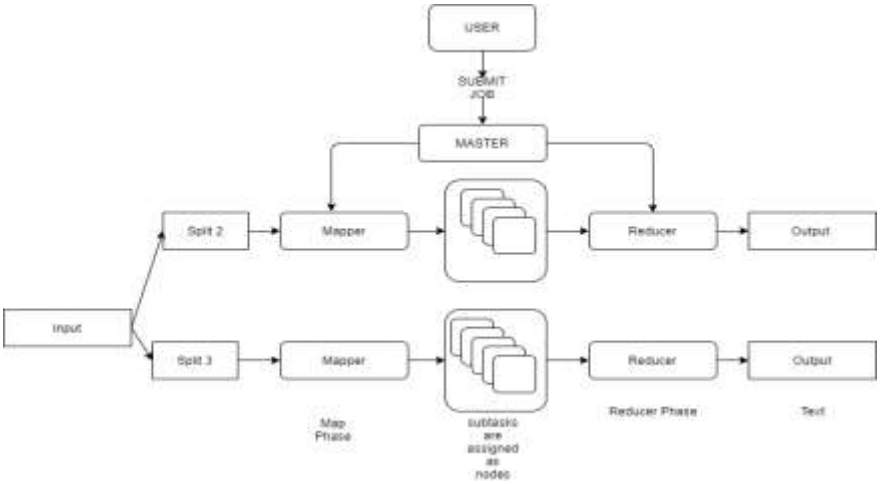

Fig 4: Mapreduce GridGain Process

3.4 Mars

Mars is a Mapreduce runtime system accelerated on graphical processing units.compares with cpu and gpus have an order of magnitude higher computation power and memory bandwidth.GPUs are designed as special-purpose coprocessors and their programming interfaces are familiar than those of mapreduce programmers.Mars was developed to run Nvidia Gpus, AMD Gpus as well as multi-core gpus. Mars hides the programming complexity of GPUs behind the simple and familiar MapReduce interface, and automatically manages task partitioning, data distribution, and parallelization on the processors. Mars is integrated in hadoop enable Gpu Acceleration for a network of Pcs.Mars limit the Gpu compilation by Mapreduce Connectivity,task partitioning,data distribution,and parallelism on the process.Mars has three stages Map,Group and reduce.while starting the map stage the mars will preprocess the data in key/value pairs in the main memory. The the key/value pairs are taken and stores in the graphical processing unit memory, and the map stage starts the split and dispatches the input record in Gpu threads that the workload of the thread is even.Each thread then performs the User-defines MapCount function to calculate the local histogram of the number.After the runtime performs the graphical

processing unit -based Prefixsum on the local histogram and calculate and putsize will write position on each thread.after completing the input prefix sum the output buffer is allocated to device memory, each graphic processing thread will execute the map function and output the results,there will be no write conflict between concurrent threads when each thread has computed its write position before it has no contrast between other threads. In the group stage the sort-based and hash-based strategies are available for grouping for the key. When some application needs all the outputs then we use a hash-based strategy to perform additional sort within each hash bucket.In reduce stage,reduce will split dispatches each group of records with an identical key to a graphic processing unit thread. The mapreduce framework in mars enables integration of graphical processing units-accelerated on distributed environments like hadoop with least effort.

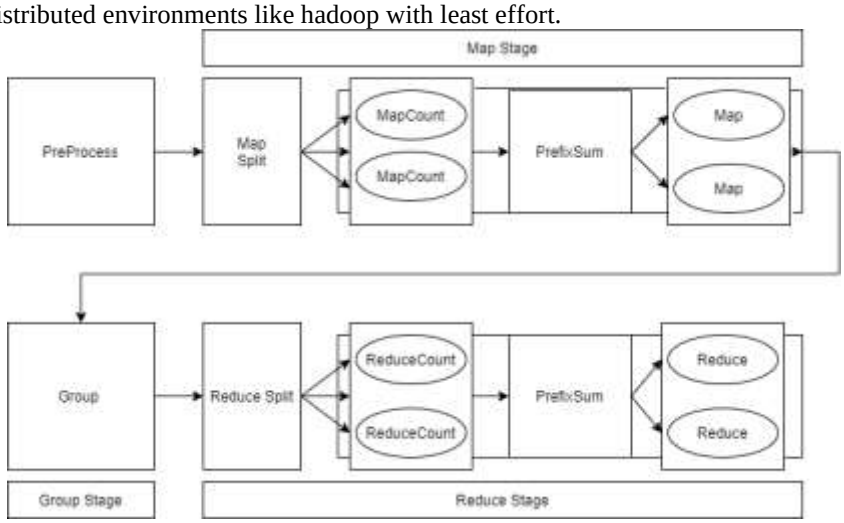

Fig 5: Mapreduce Mars Process

3.5 Phoenix

Phoenix implements shared-memory systems and it aims to support the implementation on multiple cores without troubling programmers with concurrency management.It has simple API(Application programming interface) an efficient runtime Resource management,parallelism, fault tolerance.Phoenix is finding new approaches that get over the <key,value>pairs limitation in phoenix framework using hash tables with B+trees and get over the collisions problem of hash tables.

Phoenix is an application of MapReduce.In this enhancement,it has been optimized for shared-memory multiprocessor as well as multi-cores and has the speed up with 256 threads.This enhancement also shows some roadblocks which limit the parallel runtimes scalability on shared-memory systems.However,phoenix runtime was optimized on 32 core quadchip to show the efficient run and execution at the large scale needs a special approach of multi layered optimization. In phoenix the splitter splits the input into two units.this units has the same size and this units will be processed by the map task.The Map tasks are assigned dynamically to workers.Each worker will emit the intermediate (key,value)pairs.the intermediate(key,value)pairs will be divided into units by the partition function.The reduce tasks are allocated dynamically to workers like the map tasks.But there's one difference in the map tasks we can distribute the pair across the tasks freedom,with map reduce all the values that belong to the same key must process it in one task.the output of each reduce task are sorted by a key.Merge the final output from all tasks and put them into one buffer.they must be sorted by keys.

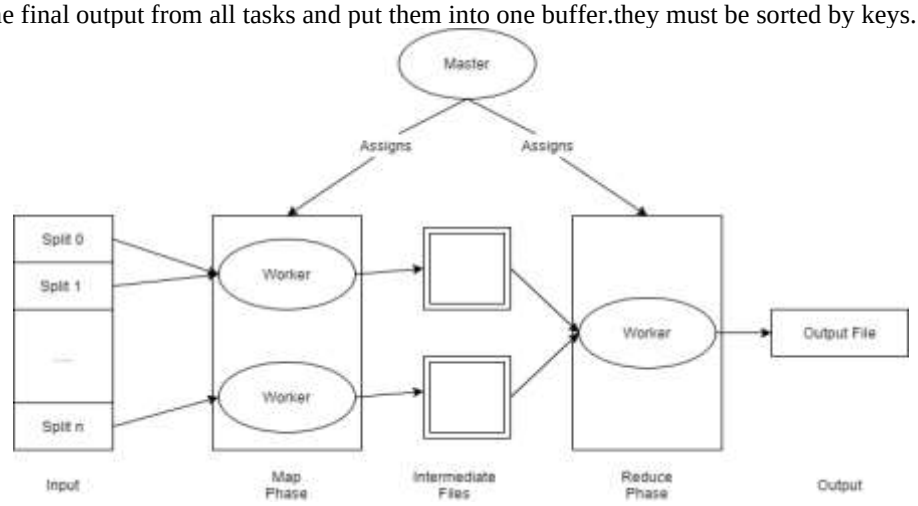

Fig 6: Mapreduce phoenix process

Table 1: Mapreduce Implementations pros and cons

Mapreduce Implementations	Pros	Cons
Hadoop	Scalability and availability are the distinguishing features to achieve data replication and fault tolerance	Cluster Management is difficult in hadoop. In the cluster, operations like debugging, distributing software, collection lags are too hard
GridGain	GridGain approach of giving tasks to the control of sub-task distribution,enables early and late load balancing algorithms	It doesn't have any support for non-java applications.GridGain has only one supporter that is gridgain
Mars	Mars is the first Mapreduce implementation on GPUs.Mars exploits the massive thread parallelism within GPU.	Its implementation does not employ automatic operations.it requires expensive processing kernels to coordinate output from different threads to the global memory
Phoenix	Phoenix is a well-organised implementation on the multi-core CPU.	The scalability of phoenix does not appear to be enhanced

4.MAPREDUCE APPLICATIONS

Map reduce applications can perform on many distributed applications.Mapreduce is used for parallel distribution for large cluster computing.It is a efficient distributed processing on different uses like:-Search,Clustering,log analysis,different join operations,Pattern matching and analysis on social networks and sensors data collection and distribution.

4.1Distributed Grep

Distributed Grep where it searches a plan-text data sets of lines of regular expression for finding the pattern within a large number of files.It is used in hadoop clusters to find log messages hidden within terabytes of log data.It utilizes all available CPU cores to run grep on log files in parallel,and has support for optional reduces to do post processing.

Distributed Grep consists of jobscripts which form the hadoop Streaming Job, a web app with a form to make it easy to perform searches,and a launcher typically run by cron every minute which submits jobs to the cluster.the web app shows the progress of the job and displays the results when the job has finished,with an option to download the results as a file. In the grep the input data has sent to map function, and tin the map function, in the map function the input lines are split into words.the mapper outputs the(key,value)pairs in the world and value.the reducer function will make the aggregation of the keys and outputs the results in to the file for further process.

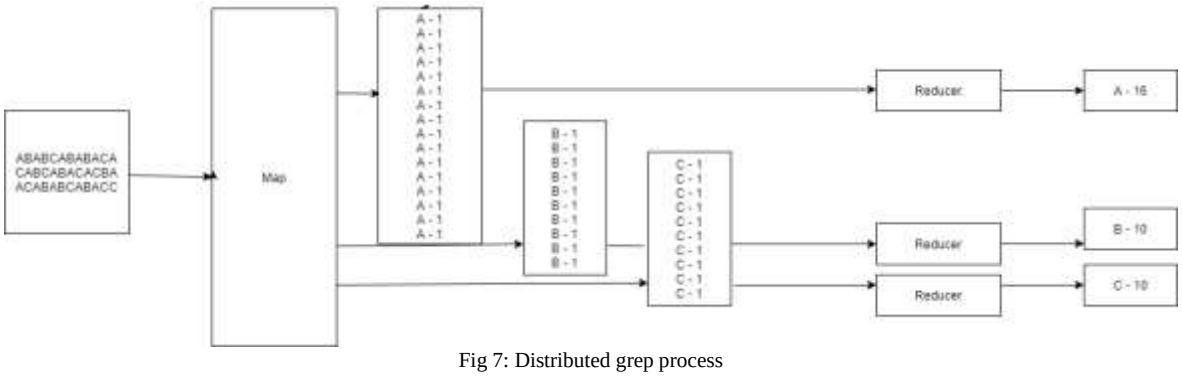

Fig 7: Distributed grep process

4.2WordCount

Wordcount is a mapreduce application which counts on occurrences of each word, enlarges text data and executes in small amounts of data from large datasets.i figure it shows how the order to count the number of words in the file;each instance of mapper receives the one line of input.IN mapper the input lines are splits into

words.mapper outputs the (key,value)pair of the word and value.since all lines are independent if each other, all mappers run parallel.

The WordCount operation takes place in three stages: Map,Shuffle,Reducer. In Mapper the input files are split into words and make alterations to key and value pairs with the words the key is being itself with '1'.For example,consider the sentence "Memory Cache Code Code Cache Code Memory Cache Code"in mapper the sentence ids split into words and from the initial key value pair as <Memory, 1> <Cache, 1> <Code, 1> <Code, 1> <Cache, 1> <code,1> <Memory, 1> <Cache, 1> <Code, 1>.After the map stage th mapper will transfer the intermediate (key,value)pairs through the shuffle pairs.the data transfer by the mapper to shuffle phase stores in mapper disks rather than main memories and the intermediate results are sorted by the group of key pairs will same keys together. In reducer, the keys are all put together and values will the same similar key are added.The 'Memory' pairs are added and output of pairs would be <memory, 2> <Cache, 3> <Code, 4>this would give the number of occurrences of each word of input and reducer forms the aggregation of keys and shows the final result.

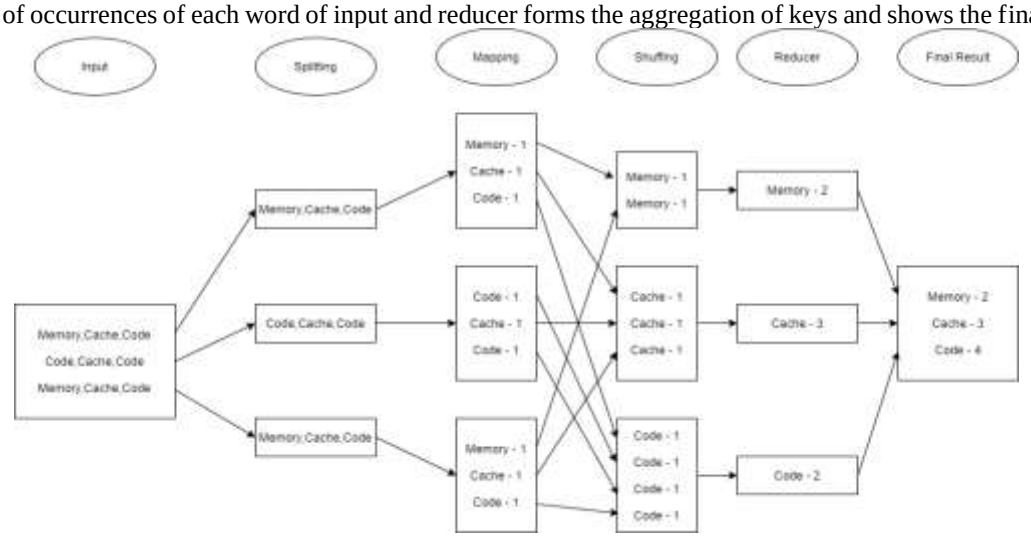

Fig 8: Mapreduce word count process

4.3Spark

Spark a cluster-computing framework supports applications with working sets providing similar scalability and fault tolerance properties to Maoreduce.it is a general-purpose processing for a wide range of circumstances.It is used to maintain scalability and fault tolerance of mapreduce by means of an abstraction called as Resilient distributed datasets(RDDs).An rdd is an read-only group of objects,which is rebuilt in a set of machines that can rebuilt if a partition is lost.an read-only collection of objects maintained in memory across iterations and supports fault tolerance.

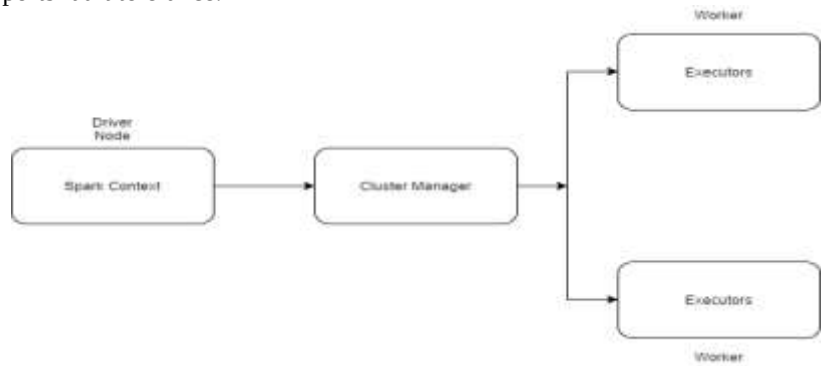

Fig 9:Spark Architecture

4.4 Mobile Sensor Data

Mobile sensors collect the data in huge ways for processing a task.Nowadays,the smartphones are widely used in technological development of embedded sensors which brings the function of sensing and communicating and computing od users.In the current situation the processing of data from the smartphone data and deploy into the private cloud.the utilization of the the Mapreduce is recognition of data by human activities for scale performance and accuracy of the process.the sensor data collected by smartphones has to match the patterns.The complication of the matching the pattern by neds of the human pattern.the data should be gathered as close as possible to real scenarios for training goals.Mapreduce are used for large collection of data by smartphones for processing of data collected by sensors

4.5 Social Networks

Social Networks collect higher amounts of data for processing and collection of data.Nowadays, it is not suitable to use traditional analysis due to rapid growth of the network.Mapreduce framework can solve large scale social network analysis problem by making use of multi-machines,which is qualified in processing large-scale social network data.Hadoop takes the place of many traditional analysis methods to conduct a series of analysis on large scale social networks,including several distributions like clustering coefficient and diameter.Nowadays the collection of data by the social network has become higher where we use the Hadoop techniques

Table 02: Mapreduce applications pros and cons

Mapreduce Applications	Pros	Cons
Distributed Grep	A generic Search tool in many Data analyses	Suffering from prolonged response time in large clusters
WordCount	Counting the occurrences of each word and massive document collection	Limiting the memory usage
Spark	Spark performs better when all the data fits in the memory,especially on dedicated clusters	Spark needs a lot of memory
Mobile Sensor Data	Help in process of the extracting context of the user like location,situation etc	Difficult to implement
Social Networks	Access large samples of respondents quickly	Develop appropriate data analytic techniques for huge samples

5. CONCLUSION

Mapreduce framework is well structured and gives many techniques to future technologies. The purpose of this essay is to explore the technology of mapreduce applications and also study several Mapreduuce implementations.the study of the technologies gives much knowledge on techniques which is used in our daily life like the mars which is used in nvidia graphics much more, this paper examines and categorized a number of applications in Mapreduce framework based on Graph processing, multi-core systems, and data allocation. The goal of the Mapreduce Framework is to provide an abstraction layer between fault tolerance,data distribution and other parallel systems tasks the mapreduce application are growing rapidly.

6. REFERENCES

[1] Jeffrey Dean and Sanjay Ghemawat: Mapreduce: Simplified Data Processing on large clusters
[2] Dean, J., Ghemawat, S.: MapReduce: a flexible data processing tool Commun. ACM 53, 72–77 (2010)
[3] Marozzo, F., Talia, D., Trunfio, P.: P2P-MapReduce: parallel data processing in dynamic Cloud environments. J. Comput. Syst. Sci. 78, 1382–1402 (2012)
[4] Dean, J., Ghemawat, S: MapReduce: simplified data processing on large clusters Commun. ACM 51, 107–113 (2008)
[5] Lu, L., Shi, X., Jin, H., Wang, Q., Yuan, D., Wu, S.: Morpho: a decoupled MapReduce framework for elastic cloud computing. Futur. Gener. Comput. Syst. 36, 80–90 (2014)