Job Schedulers In Mapreduce

Miss Jadhav Usha Dattatraya[#1], Prof V.R Chirchi[*2]
[#]*PG Student ,PG Deparment of MBES College of engg Ambajogai ,BAMU University*
[#]*Assistant Professor ,PG Deparment of MBES College of engg Ambajogai*

ABSTRACT
The aim of this paper is to provide a better understanding of job schedulers in MapReduce and identify important research directions in this area. it present advantages and disadvantages of different jobs scheduling algorithm in MapReduce also job schedulers in MapReduce and then their features and the application of each category of the schedulers are expressed and the Mapreduce is a programming model and an associated implementation for processing and generating large datasets that is amenable to a broad variety of real-world tasks users specify the computation in terms of a map and a reduce function, and the underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, handles machine failures, and schedules inter-machine communication to make efficient use of the network and disks.

Keyword:- *MapReduce Schedulers, Data Locality, Hadoop,scheduing algorithm, MapReduce.*

1.INTRODUCTION:-
MapReduce is a programming model as well as a framework that supports the model the main Idea of the MapReduce model is to hide details of parallel execution and allow users to focus only on data processing strategies[1] that users specify the computation in terms of a map and a reduce function, and the underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, handles machine failures, and schedules inter-machine communication to make efficient use of the network and disks [3] & Hadoop is an open-source implementation for MapReduce. Job scheduling in multi-user environments is an open issue that has not been well addressed yet [3] also in

Map Reduce, the job submitted by user is divided into several tasks there are two types of task in MapReduce map task and reduce task each node is a physical machine with computational and storage capabilities Hadoop uses the number of slots concept for each node in order to control the maximum number of tasks that can be executed concurrently on a node & each slot of the node at any time is only capable of executing one task. in MapReduce, there are two types of slot: map slot, and reduce slot Scheduling decisions are taken by a master node, called the JobTracker, and the worker nodes that called TaskTracker execute the tasks [2]. In section 2, it briefly discuss the job schedulers in MapReduce algorithm advantage and its drawback used for them. In Section 3, Problem Statement and Problem Defination used for job scheduler in MapReduce and MapReduce architecture , and Section 4 includes hardware and software Specification and Section 5 includes advantages and applications and conclusions.

2.LITERATURE REVIEW:-
Jeffrey Dean and Sanjay Ghemawat[3] has reviwed overall flow of MapReduce operation implementation in this MapReduce operation implementation has several advantage has the MapReduce programming model has been successfully used at google for many different purposes it attribute this success to several reasons first, the model is easy to use, even for programmers without experience with parallel and distributed systems, since it hides the details of parallelization, fault tolerance, locality optimization, and load balancing &, a large variety of problems are easily expressible as MapReduce computations & its drawback of MapReduce programing model is the completed map tasks are reexcuted on failure because their output stored on the local disk of the failed machines and is therefore inaccessible completed reduce tasks do not need to be reexcuted since their output is stored in a global file system .

Jin et.al[4] proposed the Balance Reduce algorithm which produces an initial task allocation for all map tasks allocation for all map task of a job and then takes network state and cluster workload into consideration to interactively adjust the task allocation to reduce job turnaround time & it's advantage of Balance reduce algorithm (BAR) can adjust data locality dynamically according to network state and cluster workload and the simulation result show that BAR is able to deal with large problem instances in few seconds and outperforms related algorithm in terms of job completion time & its drawback of BAR algorithm is in order to simplify BAR the authors assumed that all local map tasks spend identical execution time but this assumption is not realistic since the map task execution time even through when the processed input size is the same besides reduce task scheduling was not addressed by BAR.

Zhenhua Guo et.al[5] has investigate data locality in depth for data parallel systems among which GFS/MapReduce is representative and thus our main research target its advantage has the goodness of data

locality is a significant advantage of data parallel systems over traditional HPC system also good data locality reduces cross switch network traffic one of the bottlenecks in data intensive computing & it build a mathematical model of scheduling in MapReduce and theortically analyze the impact on data locality of computation factors such as the number of nodes and tasks it find the default Hadoop scheduling is non optimal & propose an algorithm that schedules multiple task simultaneously rather than one by one to give optimal data locality also it run extensive experiments to quantify performance improvement of our proposed algorithm such as delay scheduling algorithm measures how different factors impact data locality & investigate how data locality influence job execution time in both single-cluster &cross cluster environments.

Chen He et.al[6] has a new matchmaking algorithm is used to improve the data locality rate and the average response time of MapReduce cluster its advantages has the matchmaking algorithm carried out experiments to compare not only MapReduce scheduling algorithm but also with an existing data locality enhancement technique and the experimental results demonstrate that matchmaking algorithm can often obtain the highest average response time for map tasks & to evaluate the matchmaking scheduling algorithm it compare Hadoop default FIFO scheduler & delay scheduling algorithm in this two metrics i.e. map tasks data locality rate and average response time are used for evolution also avoid wasting computing resources the matchmaking algorithm will assign the node a non local task and the matchmaking algorithm achieves the high data locality rate & high cluster utilization and the drawback of matchmaking algorithm does not need any parameter tuning.

Zaharia et.al[7] have developed a delay scheduling algorithm to impove the data locality rate to Hadoop cluster for this the delay scheduling algorithm a jobs execution is postponed to wait for a slave node that contains the jobs input data here the delay time D is a key parameter by default it is set 1.5 times the slave nodes heartbeat interval also to obtain the best performance for delay scheduling algorithm it have to choose an appropriate D value & the drawback of delay scheduling algorithm is if the value is set too large job starvations may occur and affect performance & if the value is too small D value allows non local tasks to be assigned to fast for different kinds of workloads & hardware environments the best delay time may vary also to get an optimal delay time always needs careful tuning.

Moussa Ehsan,Radu Sion[8] introduce LiPS, a new cost-efficient data and task co-sheduler for MapReduce in a cloud environment & the advantages of LiPS is LiPS allows flexible control of job makespans , multi-resource management and fairness by using linear programming to simultaneously co-shedule data and tasks LiPS helps to achieve minimized dollar cost globally and LiPS suggest giveing priority to data placement in cloud task scheduling & it have implemented an instance of the LiPS moldel on the popular Hadoop platform & shown that significant cost saving of up to 79% can be achieved its experimental result also admit that cost reduction can often decrease the overall job execution time also LiPS is first attempt to slove the dollar cost-optimal scheduling problem & it aslo relies on the assumption that jobs and their input data files are submitted together & its drawback of LiPS is not designed to minimize individual job execution time due to the nature of the Map Reduce paradigms its cost reduction usually also results in a reduction of the total job execution time.

3.**PROBLEM STATEMENT AND PROBLEM DEFINATION** :-

Jeffrey Dean and sanjay Ghemawat[4] has implemented MapReduce progarming model that it used two functions such as map and reduce i.e. Map, written by the user, takes an input pair and produces a set of intermediate key/value pairs and the reduce function, also written by the user, accepts an intermediate key I and a set of values for that key & it merges these values together to form a possibly smaller set of values for 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 allows us to handle lists of values that are too large to fit in memory

Algorithm1 of counting the number of occurrences of each word in a large collection of documents as :-

```
map(String key, String value):
   // key: document name
   // value: document contents
   for each word w in value:
   EmitIntermediate(w, "1");
reduce(String key, Iterator values):
   // key: a word
  // values: a list of counts
   int result = 0;
  for each v in values:
  result += ParseInt(v);
  Emit(AsString(result));
```

The map function emits each word plus an associated count of occurrences & the reduce function sums together all counts emitted for a particular word.

Chen He et.al[6] has matchmaking algorithm and locality marker cleaning aigorithm is implemented therefore the matchmaking algorithm and locality marker cleaning algorithm is a MapReduce scheduling algorithm.

Algorithm2:-MatchmakingScheduling algorithm:-

```
for each node i of the N slave nodes do
   set LocalityMarker[i]=null
  end for
```

Upon receiving a heartbat from node *i*:
while node *i* has free slots, i.e., its free slot count *s>0*
set *previousMarker=LocalityMarker[i]*
for each job *j* in the *JobQueue* **do**
if job *j* has an unassigned local task *t* **then**
assign *t* to node *i*
set *s=s*-1
if *LocalityMarker[i]==null* **then**
LocalityMarker[i]=1
else *LocalityMarker[i]+=1*
end if
break for
else continue

The matchmaking algorithm is used to impove the data locality rate and the average response time of MapReduce cluster

Algorithm3 :Locality marker cleaning algorithm:-

When a new job *j* is added into the *JobQueue*:
for each node *i* of the *N* slave nodes **do**
set *LocalityMarker[i]=null*
end for

end if
end for
if *previousMarker==LocalityMarker[i]* **then**
set *LocalityMarker[i]=0* //mark this node
break while
else if *LocalityMarker[i]==0* **then**
assign node *i* a non-local task *t'* from the first job in the *JobQueue*
set *s=s*-1
break while
end if
end while

3.1Architecture of MapReduce:-

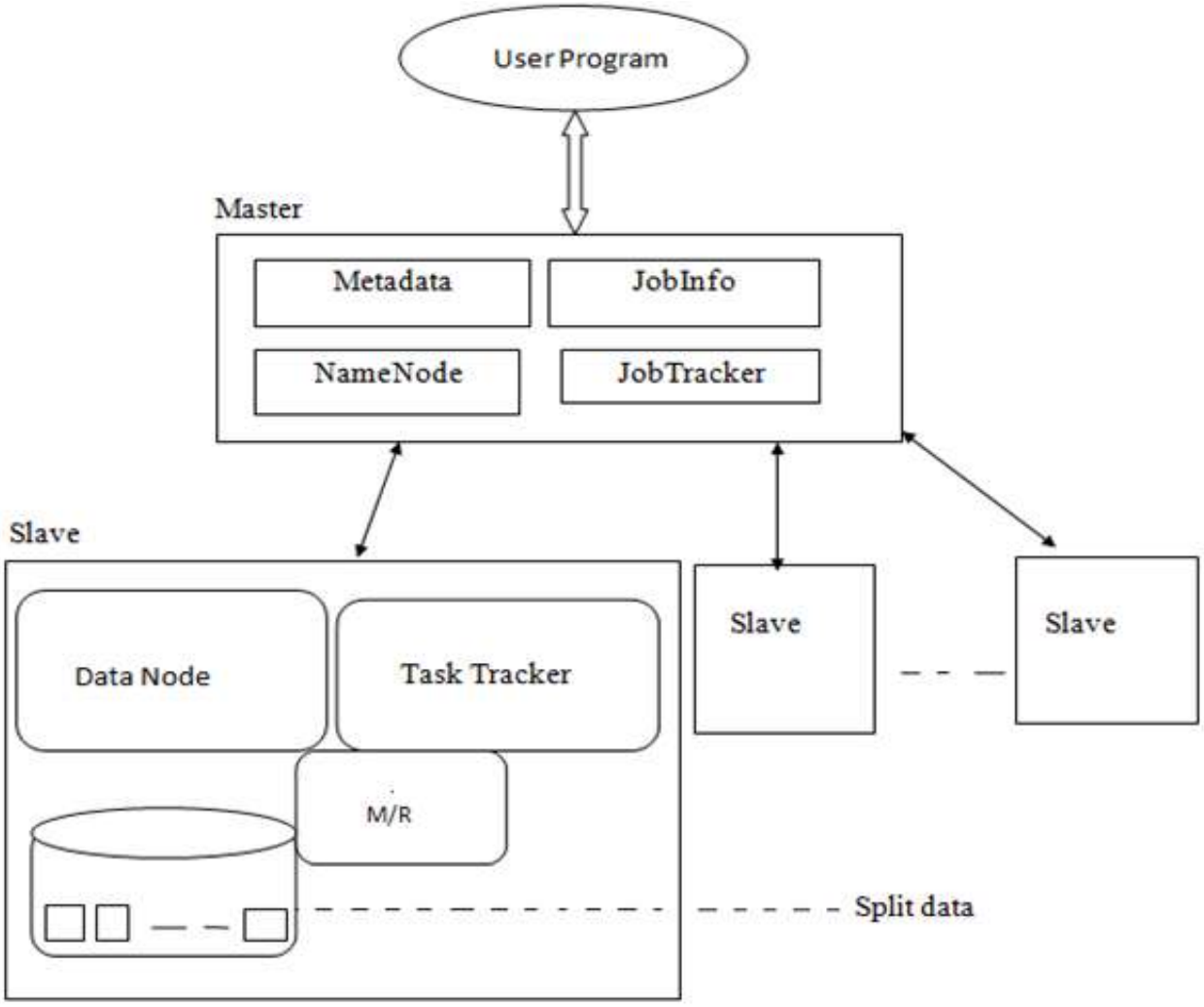

Fig1MapReduce Architecture[9]

In Fig1, the architecture of MapReduce [9] follows a traditional centralized server-client (master-slaves) model. The master node normally runs the Namenode and Jobtracker services and takes on most of the administrative work, such as high-level metadata information handling, uploaded data splitting and distribution, jobs and tasks scheduling, and command-line query handling & on the other hand, other nodes normally run with datanodes and tasktracker services, which store the real split data block and execute the

assigned tasks from master node & a general MapReduce job is spit into four different stages: Data split, Map, Shuffle, and Reduce & input data (files) hadoop book], for example, 64 MB is the default block size of Hadoop MapReduce. then, each block is stored across
datanodes according to placement assignment constructed by the master node.

Generally, the placement strategy is random and is built based on user-defined replica's factor.for example, if the replica factor is 3, Hadoop places the first split data block on a local data node within the same rack. then, it duplicates the same block to another random data node in another rack. finally, it forwards the second copy to another random data node in the same or other rack.

4. HARDWARE AND SOFTWARE SPECIFICATION

Hardware Requirements:-

Processor -Pentium –IV
Speed - 1.1 Ghz
Ram - 256 Mb
Hard Disk - 20 Gb
Key Board - StandardWindows Keyboard
Mouse - Two or Three Button Mouse
Monitor - SVGA

Software Requirements:-

Operating System : Windows XP
Coding Languag e : Java

5.ADVANTAGES OF MAPREDUCE SCHEDULING ALGORITHM :-

Balance-Reduce(BAR)algorithm schedules tasks by taking a global view and adjusts task data locality dynamically according to network state and cluster workload ,in a poor network environment, BAR tries its best to enhance data locality when cluster is overloaded, BAR decreases data locality to make tasks start early and it evaluate BAR by comparing it to other related algorithms also the simulation results show that BAR exhibit an improvement and can deal with a large problem instance in a few seconds & as a future work, it plan to implement BAR into a production cloud computing system such as Hadoop. in a real world platform, the network state and the cluster workload change frequently, so it is necessary to update the scheduling strategy by an efficient rescheduling algorithm and LiPS suggests giving jobs execution is postponed to wait for a slave node that contains the jobs input data here the delay time D is a key paremeter by default it is set 1.5 times the slave nodes heartbeat interval also to obtain the best perfomance for delay
scheduling algoritm & the LiPS allows flexible control of job makespans , multi-resource management and fairness by using linear programming to simultaneously co-shedule data and tasks LiPS helps to achieve minimized dollar cost globally and LiPS suggest giving priority to data placement in cloud task scheduling.

uploaded to GFS/HDFS are split into sequential blocks with a specified size [google paper and

Based on the selected input data format for map stage, the split data blocks are constructed into <key, value> pairs, in which key is normally represented as a unique name and is used to perform in-memory local sorting. Map functions compute the intermediate result according to assigned <key, value> pairs. The Shuffle stage sorts intermediate Key Value pairs by their keys and sends each record to an assigned reducer finally, reducers combine and compute with the collected Key Value, and yield meaningful results to the disk.

priority to data placement in cloud task scheduling & LiPS is a first attempt to solve the dollar cost-optimal scheduling problem.

Data locality is a significant advantage of data parallel systems over traditional HPC systems. Good data locality reduces cross-switch network traffic - one of the bottlenecks in data-intensive computing and the MapReduce is used for the generation of data for Google's production Web search service, for sorting, data mining, machine learning, and many other systems also it developed an implementation of MapReduce that scales to large clusters of machines comprising thousands of machines & the implementation makes efficient use of these machine resources and therefore is suitable for use on many of the large computational problems encountered at Google.

CONCLUSION:-

In this paper is introduced the MapReduce Scheduling algoritms advantages and job scheduling in MapReduce and MapReduce scheduling algorithm and MapReduce architecture & MapReduce progarming model that it used two functions such as map and reduce i.e Map, written by the user, takes an input pair and produces a set of intermediate key/value pairs and the reduce function, also written by the user, accepts an intermediate key I and a set of value pairs & the delay scheduling algorithm to impove the data locality rate to Hadoop cluster for this the delay scheduling algorithm a

The Matchmaking algorithm is used to improve the data locality rate and the average response time of MapReduce cluster & the architecture of MapReduce follows a traditional centralized server-client (master-slaves) model that the master node normally runs the Namenode and Jobtracker services and takes on most of the administrative work, such as high-level metadata information handling, uploaded data splitting and distribution, jobs and tasks scheduling, and command-line query handling.

REFERENCES

[1] K. H. Lee, Y. J. Lee et al., *"Parallel data processing with MapReduce: a survey"*, ACM SIGMOD Record, 2011, Vol. 40, No. 4, pp. 11- 20.

[2] B. T. Rao, L. S. S. Reddy, *"Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments"*, International Journal of Computer Applications, 2011, Vol. 34, No. 9,pp. 29.

[3]J. Dean, S. Ghemawat, *" MapReduce: simplified data processing on large clusters"*, Communications of the ACM, 2008, Vol. 51, No. 1, pp. 107-113.

[4] J. Jin, J. Luo, A. Song, F. Dong, and R. Xiong, *"BAR: An efficient data locality driven task scheduling algorithm for cloud computing,"* in Proc. 11th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput., May 2011, pp. 295–304.

[5] Z. Guo, G. Fox, and M. Zhou, *"Investigation of data locality in mapreduce,"* in Proc. 12th IEEE/ACM Int. Symp. uster, Cloud Grid.Comput., May 2012, pp. 419–426.

[6] C. He, Y. Lu, and D. Swanson, *"Matchmaking: A new mapreduce scheduling technique,"* in Proc. IEEE 3rd Int. Conf. Cloud Comput. Technol. Sci., Nov. 2011, pp. 40–47.

[7] M. Zaharia et al. *"Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling"* In EuroSys,2010.

[8] M. Ehsan, and R. Sion, *"LiPS: A cost-efficient data and task co-scheduler for MapReduce,"* in Proc. IEEE 27th Int. Symp. Parallel Distrib. Process. Workshops PhD Forum, May 2013,pp. 2230–2233.

[9] Tak-Lon (Stephen) Wu Computer Science, School of Informatics and Computing Indiana University, Bloomington *" MapReduce and Data Intensive Applications ".*