RESEARCHERID
THOMSON REUTERS

[Kaur* *et al.*, 6(6): June, 2017]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

NOVEL IMPROVED CAPACITY SCHEDULING ALGORITHM FOR HETEROGENEOUS HADOOP

Charanjeet Kaur[*1]& Sumanpreet Kaur[2]

[*1]Research Scholar, M.Tech (CSE), Department of CSE, CGC Technical Campus, Jhanjeri

[2]Assistant Professor (CSE), Department of CSE, CGC Technical Campus, Jhanjeri

DOI: 10.5281/zenodo.814540

ABSTRACT

For large scale parallel applications Mapreduce is a widely used programming model. Mapreduce is an important programming model for parallel applications. Hadoop is a open source which is popular for developing data based applications and hadoop is a open source implementation of Mapreduce. Mapreduce gives programming interfaces to share data based in a cluster or distributed environment. As it works in a distributed environment so it should provide efficient scheduling mechanisms for efficient work capability in distributed environment. locality and synchronization overhead are main issues in mapreduce scheduling. And it also needs to schedule multiple jobs at same time in a correct way. To solve these problems with regards to locality synchronization and fairness constrains this paper review and implements different types of scheduling methods. In this paper it implements various scheduling methods and also compares their strengths and weakness. A paper compares the performances of various schedulers and the analysis will be done over many scheduler i.e, include fair, fifo, late and capacity scheduler. Further enhancement had done on capacity scheduler.

KEYWORDS Hadoop, map reduce, cloud computing , job scheduler.

I. INTRODUCTION

A. Cloud computing

Cloud computing is a on-demand computing. The cloud user has no interaction with the physical part of the cloud. In this the multiple user can access the cloud computing services through the internet. The user have to pay only per use. The cloud computer used the distributed system platform, parallel programming model like mapreduce. Hadoop –mapreduce is a powerful computation model for processing large amount of data on distributed cluster such as cloud. It is motivated by the latest data demands.

First-in-first-out scheduler is available where the jobs are scheduled in the FIFO order. in this paper we have studied various schedulers improvement possible with hadoop and also provided some guidelines on how to improve the scheduling in hadoop

Mapreduce is now a days is widely used framework for data intensive applications. As data is increased day by day so it gives motivation to develop and use Mapreduce technology. Mapreduce provide a very useful for a easy parallel programming interfaces in a distributed environment. It is also deals with another various parallel processing of distributed node problems like scheduling ad synchronization.

It can process a large amount of data in a short time which is its most powerful advantage and feature. It also supports various another application fields like, machine learning, scientific analysis , web data analysis, astrophysics and security. Basically it is used to process a vast amount of distributed data in a short time.

The basic structure of mapreduce is based on master slave architecture. In which there are a node called master node which is responsible for managing the distribution of data on other slave nodes. The actual data is processed on all slave nodes in a distributed way. A master node monitors status of slave nodes. On a slave

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

node data is processed in two phases map and reduce phase. During the map process the intermediate results are generated and for further processing data is send to reduce or a final phase. A reduce phase receive data from map phase as a input and further process it to make a final result. In a reduce phase intermediate results are sorted using keys and then merge into a final output or result. There is another step called synchronization step is present between map and reduce step. It is basically a bridge between map ad reduce phase. A mapreduce structure is based on "share nothing" mechanism it means that every node is independent form another node. Each node know nothing about another node. It makes it easy to process data in distributed environment.

To process multiple jobs on multiple nodes using map reduce at a same time requires a efficient scheduling technique. It is essential to achieve the best performance. And it also needs to schedule multiple jobs at same time in a correct way. To solve these problems with regards to locality synchronization and fairness constrains this paper review and implements different types of scheduling methods.

There are various issues that affect the performance of scheduling techniques:

B. Scheduling issues in Mapreduce

1. **Synchronization**
2. **Locality**
3. **Fairness constrains**

1. ***Synchronization:*** It is another issue that effects the performance. Basically it's a intermediate phase of map and reduce which transfer a intermediate result from map to reduce as a input. In this all the results from map have to send to reduce and if one device is slow then all device have to wait for that device. It make the process slower. In this a single node can slow down the entire process.
2. ***Locality:*** Locality is one of the main issue of scheduling the jobs in distributed environment. It is basically a distance between input data and a node where data is assigned. That means how far inout data is from assigned device. It gives good result if data is near to the assigned device because if it nearer then it needs less amount of transfer cost .
3. ***Fairness criterion:*** A mapreduce is performed on shared and distributed cloud computing organizations like Amazon, yahoo and google. So in this some heavy workload jobs may dominate the utilization of shared nodes and another short computation jobs may not get desire time of computation. So it is important that all jobs get desire computation time for completing their jobs.

C. HADOOP

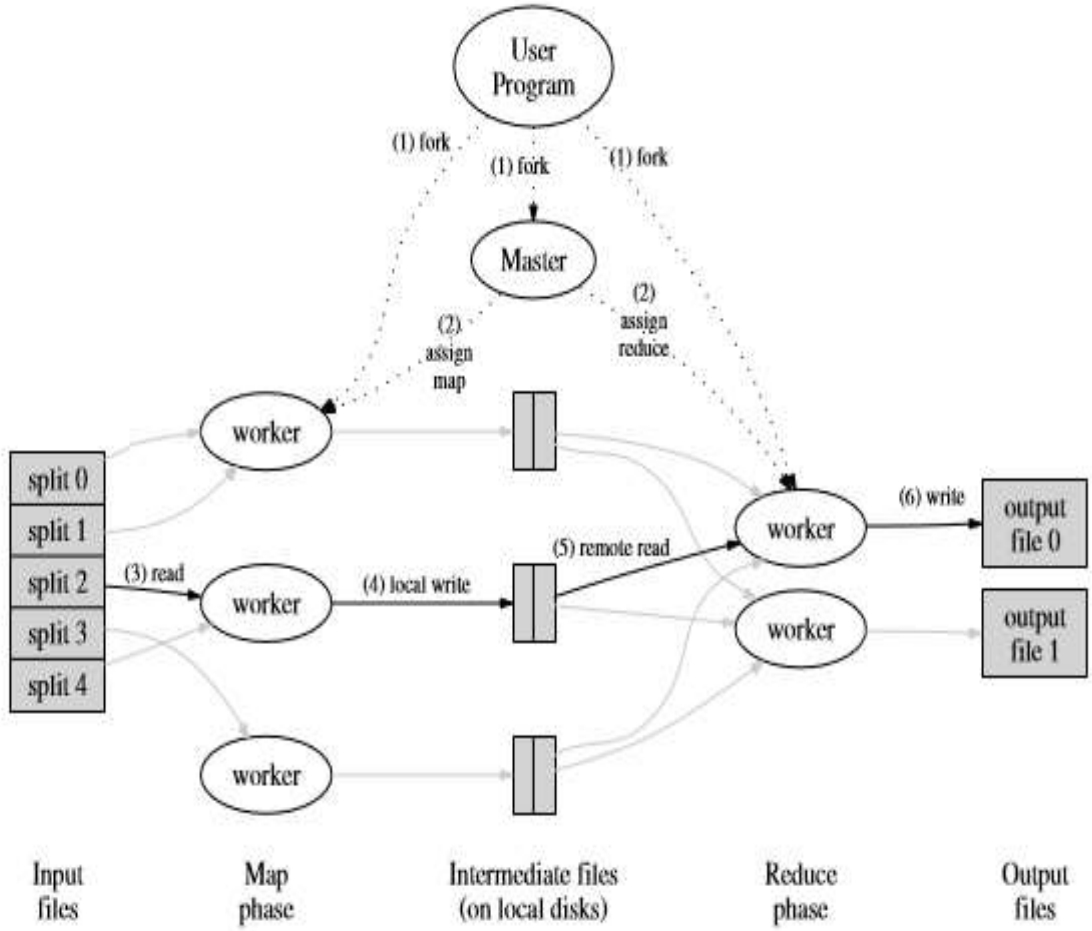
Now a days as data is increased day by day so it is essential to process and store data efficiently so that it can be used in correct way. To process and store data which has a vast amount it require a efficient technique because the traditional databases are not capable to mange such a big amount of data. Hadoop is a technique which is widely used now a days to manage large amount of data or big data. Hadoop is a architecture which process and process big data with a very efficient way.

Basically it has two parts

1. HDFS
2. Mapreduce

1. ***HDFS***: It is basically used to access the data reliably and efficiently. Hadoop distributed file system is designed and developed by google to access vast amount of data. In this data is stored into a distributed environment where data is divided into a o of junks of 64 bits. Data is stored on different large clusters. It also provide data reputation where same data set is stored o a different nodes which are located on different sites.
2. ***Mapreduce****:* Mapreduce provide a very useful for a easy parallel programming interfaces in a distributed environment. It is also deals with another various parallel processing of distributed node problems like scheduling ad synchronization. The basic structure of mapreduce is based on master slave architecture. In which there are a node called master node which is responsible for managing the distribution of data on other slave nodes. The actual data is processed on all slave nodes in a distributed way.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7



D. Various Scheduling techniques in Hadoop

1. FIFO scheduler

Hadoop scheduler operates the FIFO queue by default. Into the job partitioning, it is partitioned into individual tasks that are assigned to free slots as they available on the Task-Tracker nodes. There is a support for the priorities of jobs that is not turned on by default.

2. Fair scheduler

Fair scheduler was developed at the facebook to manage the hadoop scheduler. Fair scheduler is a method that is used to assign the resources to jobs and the resources cab be shared among multiple jobs. It organizes the jobs into pools, and divided resources fairly between these pools.

3. Capacity scheduler

It was developed at yahoo addressed and uses the scenario where the large amount of users. It allocates the jobs based on the queues that are submitted by user with configuration numbers of Map & reduce slots. Overall this has an enforcing effect to cluster capacity shared among users rather than among jobs.

4. Late Scheduler

Late Scheduler is used to robustly improve performance by reducing overhead of speculation execution tasks. This scheduler find real slow tasks by computing remaining time of all the tasks.

II. LITERATURE REVIEW:

Cloud [1] This paper presents the cloud computing design, implementation, and evaluation of a new system for on-demand provisioning . The Hadoop provide clusters are created “on-demand” and are composed of virtual machines from multiple cloud sites that are connected with bandwidth network pipes. Cloud Computing is

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

emerging as a new computational paradigm shift. Hadoop-Map Reduce has become a powerful Computation Model for processing large data on distributed commodity hardware clusters such as Clouds. In all Hadoop implementations, the default FIFO scheduler is available where jobs are scheduled in FIFO order with support for other priority based schedulers also. In this paper we study various scheduler improvements possible with Hadoop and also provided some guidelines on how to improve the scheduling in Hadoop in Cloud Environments.

Weikuan Yu, et.al [4] describe Hadoop-A, an acceleration framework that optimizes Hadoop with plug-in components for fast data movement. It overcome the existing limitations. A novel network-levitated merge algorithm is introduced to merge data without repetition and disk access. In addition, a full pipeline is designed to overlap the shuffle, merge, and reduce phases. Our experimental results show that Hadoop-A significantly speeds up data movement in MapReduce and doubles the throughput of Hadoop. In addition, Hadoop-A significantly reduces disk accesses caused by intermediate data.

Aysan et al [10]:proposed a hybrid approach that offers the use of scheduling algorithm for specific situation. Hybrid approach mainly considered average completion time for submitted job as the main performance metric. The performance can observed by using Hadoop schedulers including FIFO and fair sharing and compare it with COSHH(Classification and Optimization based Scheduler for Heterogeneous hadoop).The selection of effective scheduler is based on the load on the system and available system resources.

Scheduling techniques[1] in this define scheduling techniques fair, fifo and capacity scheduling. The Scheduling algorithm is based on FIFO the jobs were executed in the order of FIRST-IN-FIRST-OUT. The ability to set the priority of a Job was added. Facebook and Yahoo contributed significant work in developing schedulers i.e. Fair Scheduler [5] and Capacity Scheduler [6] respectively which subsequently released to Hadoop Community.

Kurazumi[13]: In this paper, they propose dynamic processing slots scheduling for I/O intensive jobs of Hadoop MapReduce focusing on I/O wait during execution of jobs. Assigning more tasks to added free slots when CPU resources with the high rate of I/O wait have been detected on each active TaskTracker node leads to the improvement of CPU performance.They have implemented it on Hadoop 1.0.3. They have evaluated up_ioline and down_ioline, there is just the little difference between the results of the values which was close. They have also concluded that setting up_ioline to the extremely high values can cause the performance decrement because the high rate of I/O wait has appeared to a mound.

They use Sort et al [14] program as benchmark program because Sort is basic operation as the program working on Hadoop MapReduce. The Map function Identity Mapper and the Reduce function Identity Reducer only get Key-Value pairs from Record Reader and output them. The amount of CPU processing in both of Map and Reduce phases is less than the amount of I/O processing, so that Sort is I/O intensive program.

GuanghuiXu, et.al [11] introduces some technologies used such as CloudStack, MapReduce and Hadoop. Based on that , a method to deploy CloudStack is given. Then we discuss how to deploy Hadoop in virtual machines which can be obtained from CloudStack by some means, then an algorithm to solve the problem that all the virtual machines which are created by CloudStack using same template have a same hostname. After that we run some Hadoop programs under the virtual cluster, which shows that it is feasible to deploying Hadoop in this way. Then some methods to optimize Hadoop in virtual machines are discussed. From this paper, readers can follow it to set up their own Hadoop experimental environment and capture the current status and trend of optimizing Hadoop in virtual environment.

III. PROPOSED WORK

The hadoop scheduler is proposed to increase the performance of the system. When the execution time is decreased the performance will increase automatically. The hadoop scheduler evaluates the good performance due to the free task assignment.

Along with if else reduce the execution cost of the scheduler. The analysis will be over many scheduler like FAIR, FIFO, LATE and as well as capacity scheduler. In the very simple & general term we can say that hadoop scheduler is used to increased the performance of the entire system.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IV. EXISTING SYSTEM

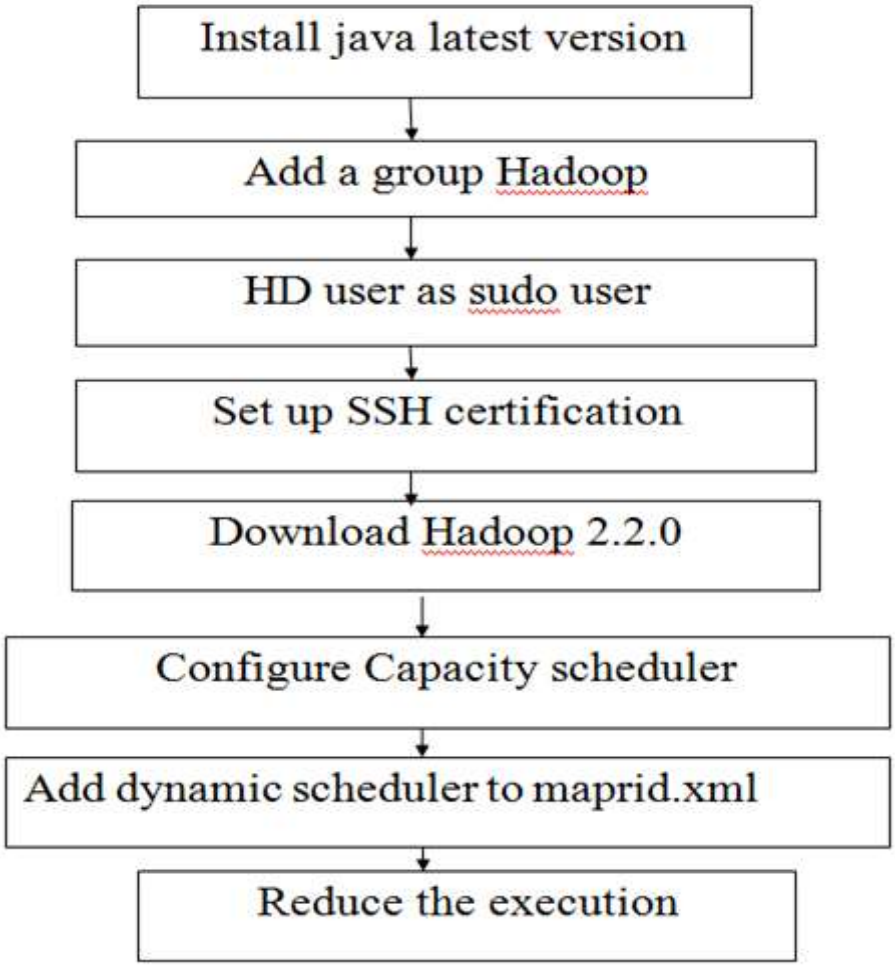
In existing work to handle the large scale data and to process it different numbers of slots are being used by users. The performance of data transmission depends on the number of slots used. When large scale data come to Job Tracker it decides which input is to given to which slot according to the amount of data it is processed to different types of slots to process the input data assigned by Job Tracker.

As the input data keep increasing on Job Tracker which in result increase the load on it. So in existing work user increased no of slots to perform the task of processing input data.

The whole execution time of the hadoop jobs using CPU resources effectively on the slave node is not up to the mark.

Assigning Task to the free slots is done statically and less was evaluated performance By increasing the number of slots performance of system was increased but number of slots are also increased which makes system costly and little heavy.

V. METHODOLOGY



Algorithm Of Improved Capacity Scheduler

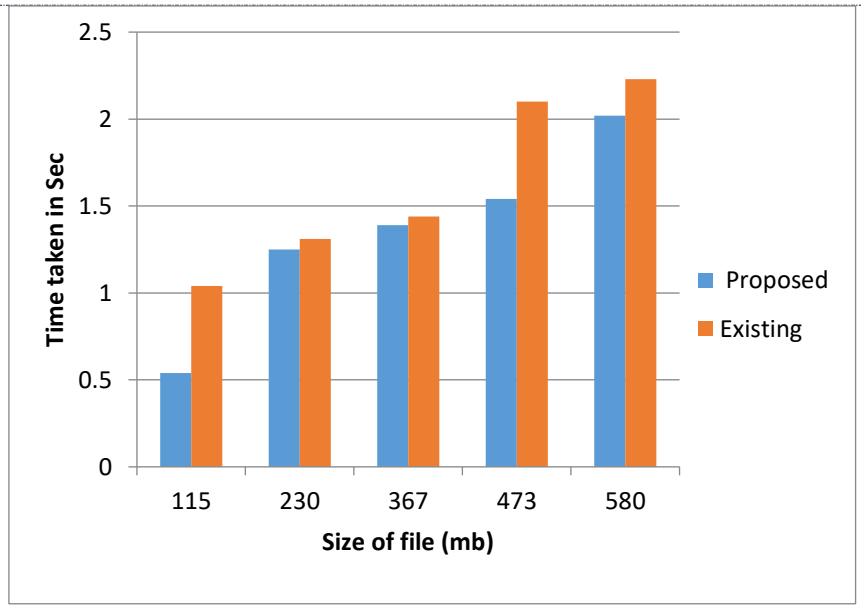
1 Initialize from Config file
2 Minimum allocation, maximum allocation, node locality, threshold
3 Initialize queues
4 For (Csleaf Q: queueManager)
5 {

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

```
6 Resource. add (resTopreempt)
7 If (Resource greater than Resource Calculator)
8 Preempt + resource (quemagr. Get LeafQueue)
// collect running container
9 for (sched: scheds)
10 {
11 If (Resource. greater than Resource Calculator)
12 {
13 for (appsched: sched.getAppSchedulable)
14 {
15 For (RMContainer: getLiveContainer)
16 {
17 Running container.add(c)
18 Apps.put (c, appsched.getApp ()
19 Queues.put (c, sched)
20 }
21 }
22  }
//kill container
25 If (time! = null)
26  {
27 If (time + waitTimeBeforeKill<clock.getTime ())
28 Create preemped container Status (container .getContainerId ())
29 CompletedContainer (containers, status, RMcontainer.KILL)
30  }
31  }
```

Size of File	Time Taken to process (Proposed)	Time Taken to process (Existing)
115 mb	0.54sec	1.04sec
230 mb	1.25sec	1.31sec
367 mb	1.39sec	1.44sec
473 mb	1.54sec	2.1sec
580 mb	2.02sec	2.23sec

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7



VI. IMPLEMENTATION

- **Installation of java**

We installed latest version of java. we installed jdk 7 of version 1.7.0_75 in our system. Java is used to write a mapreduce program that used for hadoop streaming. In hadoop ,file system used java input-output for interfacing with datainput stream and data output stream for input output operations.

Configuring the secure shell which is used to manage the nodes. It is password-less ssh so that master node can start the daemons processes on each slave node.if it is password-less ssh then job tracker should be able to send a task to task tracker quickly if not then we need to go on each individual machine and start all the processes

- **Hadoop Installation**

Hadoop is installed in our system successfully. In our implementation we worked on hadoop 2.2.0 version. For installation we extract the jar file of hadoop.

- **Configuration**

We make a NameNode and DataNode directory for heartbeat communication.

We start the distributed file system shell after formatting the hadoop distributed file system namenode. When we start the dfs services among them namenode and datanode and secondary namenode were also get start.

We start the yet another resource negotiators that provide a daemon. It also handles and schedules the resource request

The jps command which tells that whether the jobtracker, tasktracker, namenode datanode, secondary node were running succefully or not.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

```
hduser@shivani-Inspiron-1464:~$ jps
5777 SecondaryNameNode
5313 NameNode
6241 Jps
5947 ResourceManager
6157 NodeManager
5516 DataNode
```

Figure : Execution of a program

- **Execution of a program**

Before execution command we need to make a directory of a input text file and then we copy the input text file from local to hadoop distributed file system and then execute the task to count the word .
The concatenation command in which, r symbolize whether the job was a map only job or reduce job and 00000 symbolize the mapper or reduce task

VII. RESULT AND COMPARISION

In hadoop, the efficiency of the process is evaluated by execution time in different file .

- **Comparison of FIFO and capacity scheduler**

It can be seen in the table 7.1, that default scheduler is giving higher execution time than capacity scheduler

Table 7.1 : Execution time of FIFO and Capacity for different files

Scheduler	100mb	200mb	300mb	400mb	500mb	600mb
FIFO	1	1.23	2.55	3.22	4.55	5.22
Capacity	0.88	1.02	2.09	3.00	4.00	4.9

Fig 7.1 shows the graph that the FIFO takes more time than capacity scheduler in executing the task.

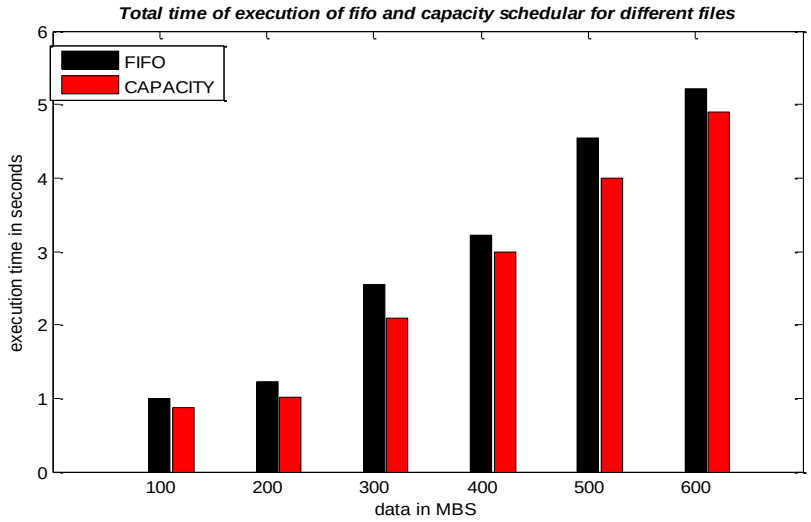


Fig 7.1 Fifo V/S Capacity

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

- **Comparison of FIFO ,Capacity and improved Capacity scheduler**

Table 7.2, show that proposed method is giving higher efficiency by decreasing the response time therefore the proposed improved scheduler is more robust than other. In figure 7.2 red bar show the improved capacity scheduler, black line show the capacity scheduler and sky blue line show the FIFO scheduler. It completely shows that proposed scheduler is best among other scheduler.

Table 7.2: Execution time over different file for different scheduler

Scheduler	100mb	200mb	300mb	400mb	500mb	600mb
FIFO	1	1.23	2.55	3.22	4.55	5.22
CAPACITY	0.88	1.02	2.09	3.00	4.00	4.9
IMPROVED CAPACITY	0.60	0.88	1.9	2.40	3.40	4.0

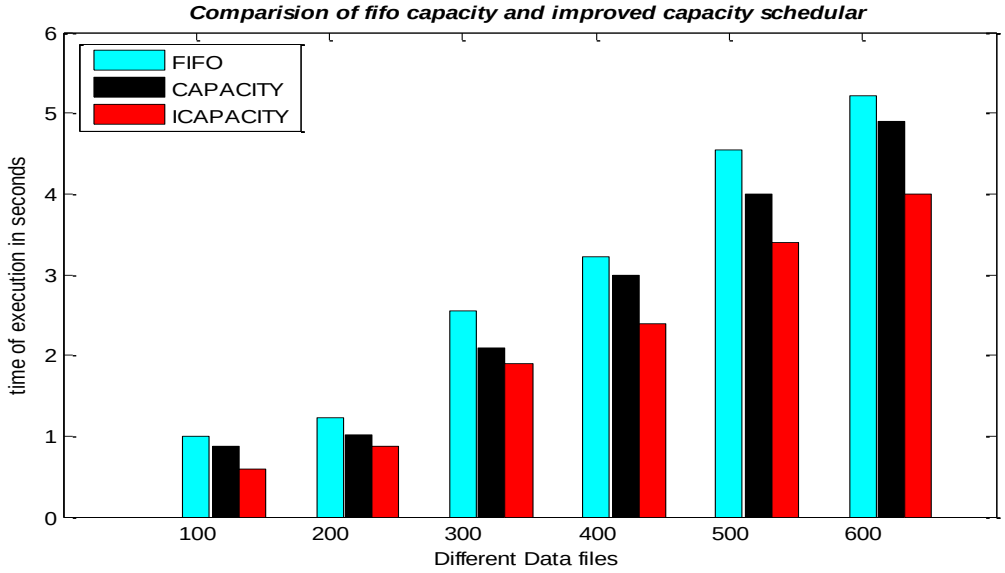


Fig 7.2 Fifo V/S Capacity V/S Improved Capacity Scheduler

VIII. CONCLUSION AND FUTURE SCOPE

- **Conclusion**

The goal of proposed algorithm during this work is to decrease the completion time of reduced tasks in map-reduce framework. in this implementation the performance of proposed algorithm is estimated from the job-completion time. We compared the proposed capacity scheduler with the default FIFO scheduler; the reduced completion time is low. We introduced queue management and piplelineing which help the task to work upon in the queue manner and shared the resources among the sub queue. And it is also proved that the average response time are decreased 29% to 50% when icapacity scheduler is applied.

- **Future Scope**

The proposed algorithm has focused on the completion time of a task by using queue management but still there is the following point that can be explored further**.**

Technique can be explored further in a heterogeneous environment for execution time and output data are consistence

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IX. REFERENCES

[1] Cloud computing, Scheduling techniques B.Thirumala Rao , Dr. L.S.S.Reddy (Volume 34– No.9, November 2011) Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments.
[2] Sagiroglu, S.; Sinanc, D. ,(20-24 May 2013),"Big Data: A Review".
[3] Apache hadoop. [Online]. Available: http://hadoop.apache.org/
[4] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," in 6th conference on Symposium on Opearting Systems Design and Implementation, Berkeley, USA, Dec. 2004
[5] Hadoop's Fair Scheduler http://hadoop.apache.org/common/docs/r0.20.2/fair_schedu ler.html.
[6] Hadoop's Capacity Scheduler: http://hadoop.apache.org/core/docs/current/capacity_sched uler.html.
[7] Dongjin Yoo, Kwang Mong Sim. A COMPARATIVE REVIEW OF JOB SCHEDULING FOR MAPREDUCE. Multi-Agent and Cloud Computing Systems Laboratory,
[8] Xu, Guanghui, FengXu, and HongxuMa(2012). "Deploying and researching Hadoop in virtual machines."Automation and Logistics (ICAL), IEEE International Conference on. IEEE
[9] Kurazumi, Shiori,(2012). "Dynamic Processing Slots Scheduling for I/O Intensive Jobs of HadoopMapReduce." ICNC
[10] Aysanet Douglas G. Down(2012)," A Hybrid scheduling approach for scalable heterogeneous Hadoop systems " SC Companion: High Performance Computing, Networking Storage and Analysis.
[11] Weikuan Yu, Yandong Wang, and XinyuQue(2014)," Design and evaluation of network-leviated merge for Hadoop Acceleration" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25.

CITE AN ARTICLE

Kaur, C., & Kaur, S. (2017). NOVEL IMPROVED CAPACITY SCHEDULING ALGORITHM FOR HETEROGENEOUS HADOOP. *INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY, 6*(6), 401-410. doi:10.5281/zenodo.814540