Optimal Scheduling algorithm for Map reduce tasks on multinode hadoop cluster

[1]Sweta Rajput, [2]Prof. Yagnesh Patel

[1]ME Scholar, [2]Assistant Professor

[1,2] Computer Engineering Department,
[1,2]S.P.B.Patel Engineering College, Linch, Mehsana, Gujarat,India

Abstract : Data generation has increased drastically over the past few years due to the rapid development of Internet-based technologies. This period has been called the big data era. Big data offer an emerging paradigm shift in data exploration and utilization. The MapReduce computational paradigm is a well-known framework and is considered the main enabler for the distributed and scalable processing of a large amount of data. However, despite recent efforts toward improving the performance of MapReduce, scheduling MapReduce jobs across multiple nodes has been considered a multi-objective optimization problem. This problem can become increasingly complex when virtualized clusters in cloud computing are used to execute a large number of tasks. Here we have presented novel Hadoop scheduler to optimize MapReduce job scheduling based on the completion time, locality, synchronization and fairness in allocation. Proposed algorithm is priority based job allocation algorithm which allocates jobs to node by considering efficiency rate of nodes.

***IndexTerms* - Hadoop Cluster, Multinode, MapReduce, scheduling.**

I. INTRODUCTION

Recently, large volumes of data or Big data have been continuously produced from daily activities, such as those involving smart phones, sensors, factories, and business transactions; these big data affect nearly every aspect of modern society [1]

Data, in today's business and technology world, is indispensable. The Big Data technologies and initiatives are rising to analyze this data for gaining insights that can help in making strategic decisions. The concept evolved at the beginning of 21st century, and every technology giant is now making use of Big Data technologies. Big Data refers to vast and voluminous data sets that may be structured or unstructured. This massive amount of data is produced every day by businesses and users. Big Data analytics is the process of examining the large data sets to underline insights and patterns. The Data analytics field in itself is vast. Some sectors are positioned for greater gains from the use of big data. Like banks, health care industries, social media sites, jobs sites they are using this big data information to make profit.

Big data is high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making.[2]

Big Data is a broad term for data sets so large or complex that they are difficult to process using traditional data processing applications. Challenges include analysis, capture, curation, search, sharing, storage, transfer, visualization, and information privacy.[2]

- **Volume :-**
  The quantity of generated and stored data. The size of the data determines the value and potential insight, and whether it can be considered big data or not.
- **Variety:-**
  The type and nature of the data. This helps people who analyze it to effectivelyuse the resulting insight. Big data draws from text, images, audio, video; plus it completes missing pieces through data fusion.
- **Velocity**
  In this context, the speed at which the data is generated and processed to meet the demands and challenges that lie in the path of growth and development. Big data is often available in real-time.
- **Veracity**
  The data quality of captured data can vary greatly, affecting the accurate analysis.[3]

II. LITERATURE REVIEW

Millions of users share cloud resources by submitting their computing task to the cloud system. Scheduling these millions of task is a challenge to Cloud Computing environment. Scheduling over comes the problems between the user and resources. When the number of users in the cloud gets increased then the scheduling of the requests of the users becomes a critical task. Therefore there is a need to go for a better scheduling algorithm than the existing one. This can be done by comparing and evaluating the various existing algorithms, there by identifying the gaps in the existing algorithms. The proposed methodology mainly consider the Max min strategy for scheduling and tries to overcome the drawbacks of existing scheduling algorithms.

This section presents the literature review work carried out in load balancing algorithm in cloud computing system. With the help of this literature review work more efficient load balancing algorithm is developed and is presented in proposed work section.

S.R.	Paper Title	Author	Journal	Year
1	An Enhanced Task Scheduling in CloudComputing Based on Hybrid Approach.	Mokhtar A. Alworafi, AtyafDhari, Sheren A. El-Booz, Aida A. Nasr, Adela Arpitha and SureshaMallappa	Data Analytics and Learning, Springer.	2018
2	Load Balancing in Cloud ComputingUsing Modified Throttled Algorithm [16]	Domanal S.G., Reddy G.R.M.	International Conference on Cloud Computing in Emerging Markets (CCEM)- IEEE	2014
3	Optimal Load Balancing in Cloud ComputingBy Efficient Utilization of Virtual Machines [18]	Domanal S.G., Reddy G.R.M.	Sixth International Conference on Communication Systems and Networks (COMSNETS)-IEEE	2014
4	A Novel Approach for Load Balancing in Cloud Data Center [20]	GulshanSoni, Mala Kalra	International Advance Computing Conference-IEEE	2014
5	LoadBalancing On Cloud Data Centers [17]	Hemant S. Mahalle, Parag R. Kaveri, VinayChavan	International Journal of Advanced Research in Computer Science and Software Engineering	2013

III. IMPLEMENTATION

Internet applications are accessed by users all around the world, and popularity and experience in use of these applications varies region by region along the world. Cloud Dataproc is a fast, easy-to-use, fully-managed cloud service for running clusters in a simpler, more cost-efficient way. Operations that used to take hours or days take seconds or minutes instead, and you pay only for the resources you use (with per-second billing). Cloud Dataproc also easily integrates with other Google Cloud Platform (GCP) services, giving you a powerful and complete platform for data processing, analytics and machine learning.

Proposed Algorithm:

1. Calculate efficiency rate of each data node and store it in E_{ri};
2. Create proposed scheduler.xml
3. Configure it with yarn-site.xml
4. Set the properties and values
5. Configure hadoop class path to reset the path for updated scheduler

//For allocation, calculate number for the same type of VM;

6. CalculateNumberofVM;
7. datanode=NumberofVM-1;
8. calculate E_{ri} for all VMs which works as datanode
9. Efficiency rate (E_r) =(Total time by map task +reduce task / Total No. of map task +reduce task) * 100%
10. Take job as input
11. Assign job to the most efficient node to get best result.
12. If you are considering priority than assign job to node having highest E_r
13. calculate execution time;
14. end

Steps to use proposed scheduler

Step 1: To use the Proposed Scheduler first assign the appropriate scheduler class in yarn-site.xml:

```
<property>
<name>yarn.resourcemanager.scheduler.class</name>
<value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.proposed.Proposed-Scheduler</value>
</property>
```

Step 2:
Set the scheduler-wide options by adding configuration properties in the yarn-site.xml file in your existing configuration directory
Step 3:
Reset Hadoop classpath
Step 4:
Set Dataproc properties
Step 5:
Now run word count program and check the timings

IV. RESULT ANALYSIS

In google dataproc, 5 node hadoop cluster was created. It was compared with inbuilt FAIR scheduler and Capacity scheduler.

Following are the tables to compare total time taken by map task and total time taken by reduced task.

NODE NAME	FILE SIZE	CPU time spent (ms)	TOTAL TIME SPENT BY ALL MAPS IN OCCUPIED SLOTS (ms)	TOTAL TIME SPENT BY ALL REDUCES IN OCCUPIED SLOTS (ms)	TOTAL TIME SPENT BY ALL MAP TASKS (ms)	TOTAL TIME SPENT BY ALL REDUCE TASKS (ms)
Sweta-m	100 MB	26700	116340	277096	29085	34637
Sweta-w-0	200 MB	58510	529308	205408	132327	25676
Sweta-w-2	400 MB	143720	1953780	219056	488445	27382
Sweta-w-3	600 MB	204290	2306616	207072	576654	25884
Sweta-w-1	800 MB	251220	3507184	207992	876796	25999
Sweta-w-4	1 GB	296450	3536500	213128	884125	26641

Table 4.1: Assign total time using Capacity Scheduler

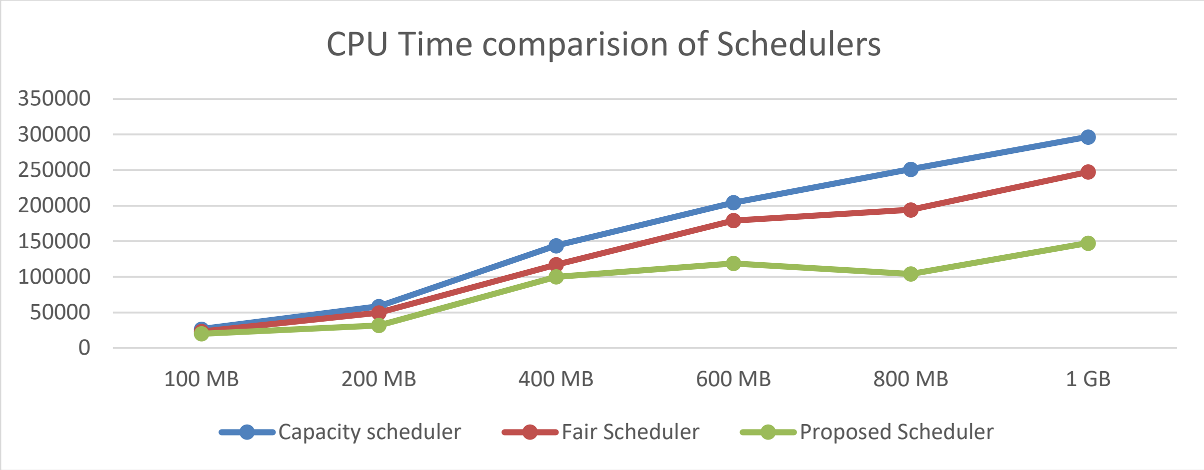
NODE NAME	FILE SIZE	CPU time spent (ms)	TOTAL TIME SPENT BY ALL MAPS IN OCCUPIED SLOTS (ms)	TOTAL TIME SPENT BY ALL REDUCES IN OCCUPIED SLOTS (ms)	TOTAL TIME SPENT BY ALL MAP TASKS (ms)	TOTAL TIME SPENT BY ALL REDUCE TASKS (ms)
Sweta-m	100 MB	23340	102024	242784	25506	30348
Sweta-w-0	200 MB	49610	403388	194816	100847	24352
Sweta-w-2	400 MB	117170	2324988	175752	581247	21969
Sweta-w-3	600 MB	178970	2737972	174832	684493	21854
Sweta-w-1	800 MB	194160	2907012	171040	726753	21380
Sweta-w-4	1 GB	247380	3239000	172752	809750	21594

Table 4.2: Assign total time using FAIR Scheduler

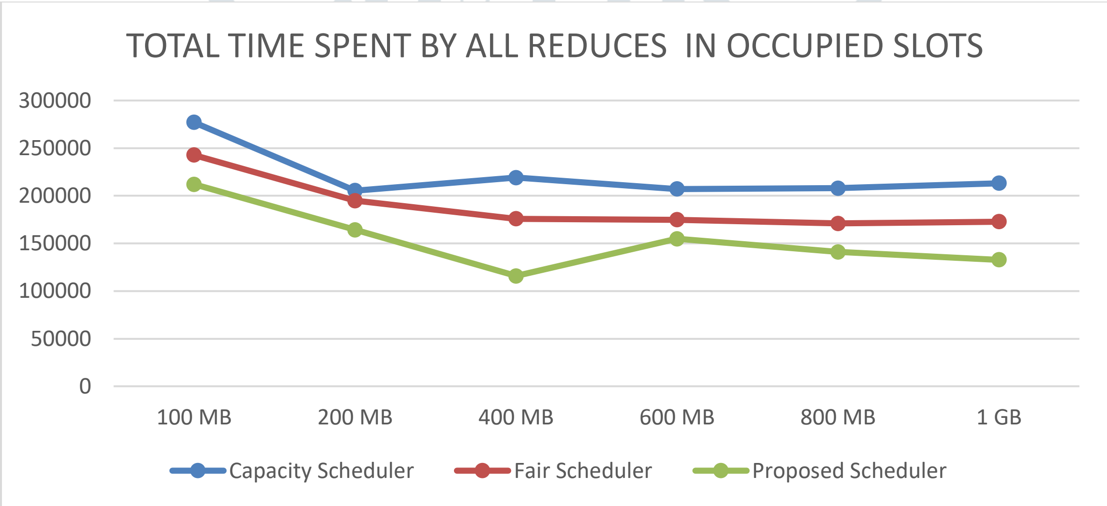
NODE NAME	FILE SIZE	CPU time spent (ms)	TOTAL TIME SPENT BY ALL MAPS IN OCCUPIED SLOTS (ms)	TOTAL TIME SPENT BY ALL REDUCES IN OCCUPIED SLOTS (ms)	TOTAL TIME SPENT BY ALL MAP TASKS (ms)	TOTAL TIME SPENT BY ALL REDUCE TASKS (ms)
Sweta-m	100 MB	20140	100000	212004	15001	10249
Sweta-w-0	200 MB	31690	358388	164237	96877	23155
Sweta-w-2	400 MB	100110	2024389	115758	511243	17969
Sweta-w-3	600 MB	118938	2537870	154841	614381	18854
Sweta-w-1	800 MB	104159	2107019	141041	656750	20381
Sweta-w-4	1 GB	147484	1239023	132743	719755	19593

Table 4.3Assign total time using Proposed Scheduler

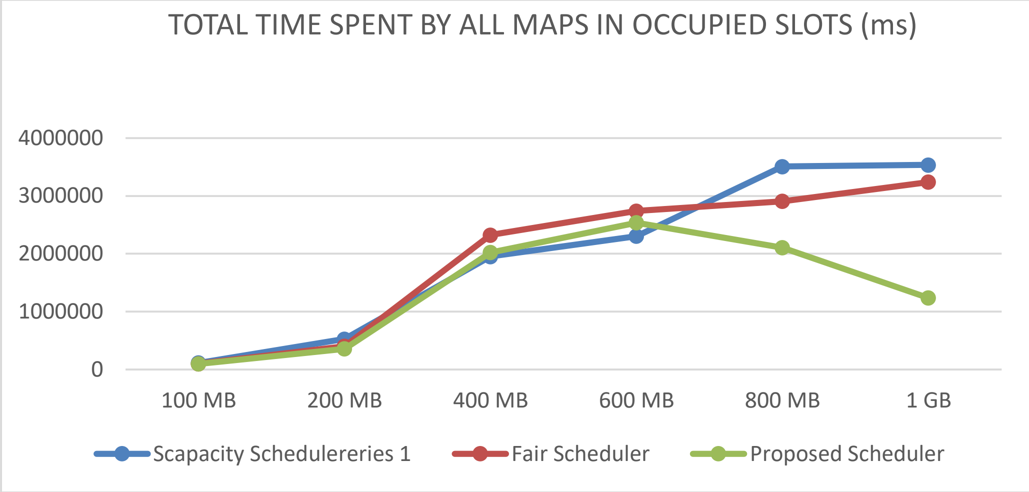
File Size	CPU Time in milliseconds		
	Capacity	Fair	Proposed
100 MB	26700	23340	20140
200 MB	58510	49610	31690
400 MB	143720	117170	100110
600 MB	204290	178970	118970
800 MB	251220	194160	104160
1 GB	296450	247380	147380



File Size	TOTAL TIME SPENT BY ALL REDUCES IN OCCUPIED SLOTS (ms)		
	Capacity	Fair	Proposed
100 MB	277096	242784	212004
200 MB	205408	194816	164237
400 MB	219056	175752	115758
600 MB	207072	174832	154841
800 MB	207992	171040	141041
1 GB	213128	172752	132743



File Size	TOTAL TIME SPENT BY ALL MAPS IN OCCUPIED SLOTS (ms)		
	Capacity	Fair	Proposed
100 MB	116340	102024	100000
200 MB	529308	403388	358388
400 MB	1953780	2324988	2024389
600 MB	2306616	2737972	2537870
800 MB	3507184	2907012	2107019
1 GB	3536500	3239000	1239023



File Size	TOTAL TIME SPENT BY ALL MAP TASKS (ms)		
	Capacity	Fair	Proposed
100 MB	29085	25506	15001
200 MB	132327	100847	96877
400 MB	488445	581247	511243
600 MB	576654	684493	614381
800 MB	876796	726753	656750
1 GB	884125	809750	719755

File Size	TOTAL TIME SPENT BY ALL REDUCE TASKS (ms)		
	Capacity	Fair	Proposed
100 MB	34637	30348	10249
200 MB	25676	24352	23155
400 MB	27382	21969	17969
600 MB	25884	21854	18854
800 MB	25999	21380	20381
1 GB	26641	21594	19593

V. Conclusion and Future Work

So it can be concluded that proposed scheduler works better than inbuilt capacity scheduler and FAIR scheduler. It gives better performance in less time. It shows that proposed model selects best node to work first which in total increases its overall efficiency. As per result it is clear that map task and reduce task time reduces in presence of proposed scheduler.

We have not considered machine learning algorithm. In future we can incorporate machine learning algorithm with it.

REFERENCES

[1] Hadoop A (2009) Fair Scheduler https://hadoop.apache.org/docs/stable1/fair_scheduler.html

[2] Heintz B, Chandra A, Sitaraman RK (2012) Optimizing mapreduce for highly distributed environments.arXiv preprint arXiv:1207.7055

[3] Hashem IAT, Yaqoob I, Anuar NB, Mokhtar S, Gani A, Khan SU (2015) The rise of Bbig data^ on cloudcomputing: review and open research issues. Inf Syst 47:98–115

[4] Huang S, Huang J, Dai J, Xie T, Huang B (2011) The HiBench benchmark suite: characterization of theMapReduce-based data analysis. New Frontiers in Information and Software as Services,Springer, pp 209–228

[5] Hussain H, Malik SUR, Hameed A, Khan SU, Bickler G, Min-Allah N et al (2013) A survey on resourceallocation in high performance distributed computing systems. Parallel Comput 39(11):709–736

[6] Ibrahim S, Jin H, Lu L, He B, Antoniu G, Wu S (2012) Maestro: replica-aware map scheduling formapreduce. Paper presented at the cluster, cloud and grid computing (CCGrid), 2012 12th IEEE/ACMinternational symposium on

[7] Isard M, Prabhakaran V, Currey J, Wieder U, Talwar K, Goldberg A (2009) Quincy: fair scheduling fordistributed computing clusters. Paper presented at the Proceedings of the ACM SIGOPS 22nd symposiumon operating systems principles

[8] Jagadish H (2015) Big data and science: myths and reality. Big Data Res 2(2):49–52

[9] Jiang D, Ooi BC, Shi L, Wu S (2010) The performance of mapreduce: an in-depth study. Proc VLDBEndowment 3(1–2):472–483

[10] Kc K, Anyanwu K (2010) Scheduling hadoop jobs to meet deadlines. Paper presented at the cloudcomputing Technology and science (CloudCom), 2010 I.E. Second international conference on

[11]Krish K, Anwar A, Butt AR (2014) [phi]Sched: a heterogeneity-aware Hadoop workflow scheduler. Paperpresented at the Modelling, Analysis & Simulation of computer and telecommunication systems(MASCOTS), 2014 I.E. 22nd international symposium on

[12] Laurila JK, Gatica-Perez D, Aad I, Blom J, Bornet O, Do T-M-T,. .. Miettinen M (2012) The mobile datachallenge: big data or mobile computing research. Paper presented at the Proceedings of theWorkshop on theNokiaMobile Data Challenge, in Conjunction with the 10th International Conference on Pervasive Computing

[13]Li J-J, Cui J, Wang D, Yan L, Huang Y-S (2011) Survey of MapReduce parallel programming model.Dianzi Xuebao (Acta Electron Sin) 39(11):2635–2642

[14]Long S-Q, Zhao Y-L, ChenW(2014)MORM: a multi-objective optimized replication management strategyfor cloud storage cluster. J Syst Archit 60(2):234–244

[15]Lopes RV, & Menasce D (2016) A taxonomy of job scheduling on distributed computing systems. IEEETransactions on Parallel and Distributed Systems 27(12):3412–3428