A Persuasive Resource-Aware Allocation Scheduler for Enhancing Task Scheduling

R.Rengasamy, M.Chidambaram

Abstract: The deployment of Map Reduce has been built to grant enhancements to total system objectives such as job throughput. Hence, the support for user-specific objectives and resource allocation management has been least regarded and addressed. Schedulers enable users to assign jobs to queues that fulfil shared of specific resource. Existing work mainly focus on scheduling glitch occurring on the master's side where the scheduler on the master node tries to allocate same work across all the worker nodes. The proposed scheduler focus on enhancing resource allocation when various kinds of workloads execute on the clusters. In order to evaluate the performance on the proposed scheduler which enhances resource utilization, an accomplishing time goal with each job is created.

Keywords: MapReduce, Big data, Resource allocation, Hadoop.

I. INTRODUCTION

The growth of the data is being increased exponentially day to day due to the usage of internet all over the world [1]. The data that occurs through the various resources is said to be in an unstructured format that grows 50 times faster than the structured data [2]. Unstructured data will not fit into relational tables due to its predefined model. These are generally heavy-weight data that could contain information based on temporal, spatial etc. The result is that this could be difficult for the users to understand with the traditional computer techniques [3]. The unstructured data should be converted to a structured data so that this could solve the difficulty issues in the field of computers. The main advantage of big data is that it decreases the cost of storing and computing the data. Before the invention of big data technologies, company used the process of relational data bases for storing and retrieving the data.

The unstructured data could be ignored by this traditional method. The approach to influence philosophy of big data varies from developing present enterprise data architecture to integrating big data as well as bringing business value. This technology helps us in taking real-time decisions that could increase the market price. However, the suitable tools are required to attain, systematize and gain values from philosophy of big data to capitalize one concealed relationships as well as recognize novice insights.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author
R. Rengasamy*, Research Scholar A.V.V.M Sri Pushpam College, Poondi, Thanjavur, India.
M. Chidambaram Department of Computer Science, Rajah Serfoji Government College,Thanjavur, India.

The embodiment and study of these techniques could enhance stronger production, increase cut-throat point and larger innovation [4]. Now-a-days, business enterprises and research group have observed and experienced a tremendous growth in research and development of data-analytics in an unprecedented way. The reason behind this uniqueness is the advent and implementation of the Map Reduce programming exemplar and its open-source adoption of Hadoop. In real-world situation, each job concurrently execute in the same data center. The Hadoop framework consists of single queue which is employed for job scheduling in FIFO manner. Most schedulers make use of multiple queues for allocation of several resources existing in a cluster. Given that capabilities are important for aggrandizing the deployment of data processing technologies. There is increase in employing ad-hoc for smaller tasks. Additionally in Cloud infrastructure allow clients to pay for resources employed. Thus, ensuring consistency between resource and utilization is a vital factor to the business architecture in Cloud. Resource management falls under the most vital factor as cloud providers are necessitate high levels of resource utilization and automation, thereby evading challenges prevailing in big data tools. Hadoop MapReduce is described as programming framework for executing big data by employing huge clusters of nodes. In these frameworks of distributed computing, the identical cluster is separated every user can divide the identical cluster for myriad of objectives and purposes. As a result various kinds of workloads will execute on the same data center such as a cluster may be employed for data mining from logs and processing web text which hinge on CPU capability, utilization and I/O bandwidth [5]. The functioning of a master-worker system like MapReduce framework links to its task scheduler on the master. Many researchers have been done in the field of scheduling problem. Existing work mainly focus on scheduling glitch occurring on the master's side where the scheduler on the master node tries to allocate same work across all the worker nodes. In the proposed scheduler, we focus on enhancing resource allocation when various kinds of workloads execute on the clusters. In real-world situation, each job concurrently execute in the same data center. Hence affects the throughputs well as performance in the whole system as shown in figure 1. CPU and I/O bound processing is ancillary component [5]. The writing of a task to the disk may be obstructed and it is averted from using the CPU until the I/O finishes [6]. This leads to CPU bound task that are scheduled on a machine which gets stopped by using the IO resources [7][8].

Whenever variety of workloads executes on such an environment, machines could provide different part of resource for various kinds of work [9] [10].

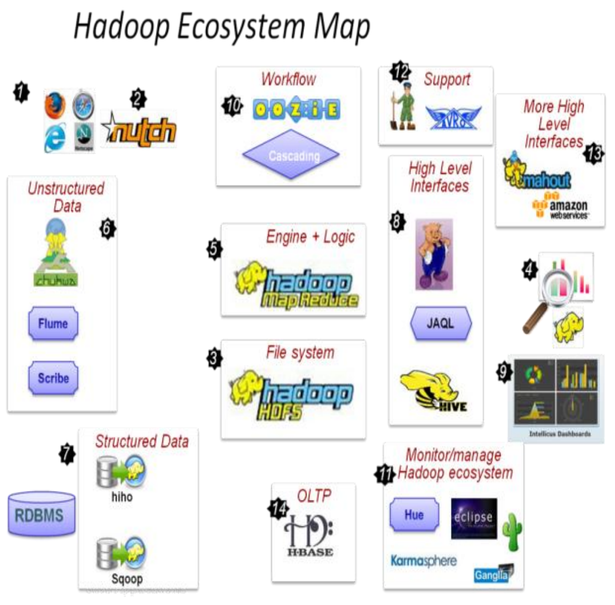

Figure 1: Hadoop Ecosystem

The rest of the paper is structured as, in section 2: problem statement is explained, in section 3: presents proposed scheduler, in section 4: presents evaluation and finally in section 5: presents future work

II PROBLEM STATEMENT

A group of MapReduce jobs $G= \{0, 1,\ldots\ldots g\}$ and a group of Task-Trackers $SS = \{0,1,\ldots..s\}$. We also state m and SS to index into the sets of jobs and Task-Trackers [11[12]. For each TaskTracker S we correlate a series of resources, $P = \{0,1,\ldots.p\}$. Every resource of Task-Tracker S contains a correlated capacity V.

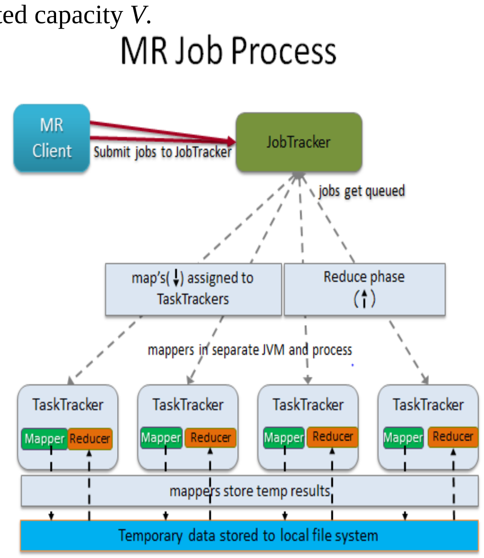

Figure 2: MapReduce Process

Considering the disk bandwidth, memory and CPU capacities for each TaskTracker and proposed algorithm is designed to contain other resources such as storage capacity.

A MapReduce job, (m) contains a group of tasks, called as offering time as shown in figure 2, each job-slot is given a specific job, and the scheduler will be managing the number of job-slots to build on each TaskTracker for each job in the scheduler. Each job j is correlated with an accomplishing time aim, S^m_{aim}, i.e. time at which the job should be finished. If accomplishing aim is not provided, then it is presumed that the job should be over at the initial likely time. Each job is provided with resource consumption parameter.

III PROPOSED SCHEDULER

A. *Architecture*

In the proposed, every TaskTracker node observers such as disk channel IO, CPU utilization. These are three fundamental metrics that will be monitored in order to enhance load balancing. Disk channel filling can bring important effect on loading of data, writing of Map and Reduce tasks, hence free memory is available, similarly the intrinsic capacity of a machine's virtual memory. Tracking these metrics leads to better management of page faults, virtual memory-induced and free the memory. Every resource request contains tuple of values such as one value combined for each task type. In our scheduler, we hold a fixed maximum number of slots per node, regarding it as a resource allocation result created by the cluster manager at configuration time. The organization of free TaskTracker depends on their availability of resource. Whenever TaskTracker slots become available, they are stalled for some period of time and later broad-casted in a block. TaskTracker slots which consists higher resource availability are considered as top priority for scheduling.Unlike scheduling a task onto the succeeding available free slot, job response time will be enhanced by scheduling it onto a resource-affluent machine, even though a node consumes much time to become free and available. Resource management falls under the most vital factor as cloud providers are necessitate high levels of resource utilization and automation, thereby evading challenges prevailing in big data tools.

B. Allocation Algorithm

The aim of the proposed allocation algorithm is to discover and allocate jobs on Task-Tracker, thereby enhancing resource utilization.

Notations Used in Allocation Algorithm:

L^M(job,S) – Map Tasks

D^R(job,S)- Reduce Tasks

S- Group of Task-Trackers

G- Group of Jobs

The inputs are L^M(job, S), D^R(job, S), S and G. The algorithm proceeds as follows:

Allocating Mappers

1. for job residing in G do
2. sort S in increasing order
   number of map tasks and
   jobs located
3. for ss in S do

If space for fresh job slot (job, S)
then
L^M(job, S)= L^M(job, S) +1
4. end if
5. end for
6. *end for*

Allocating Reducers

1. for round=1……rounds do
2. for ss in S do
3. $job_{arrival}$= min U($job_{arrival}$, D),
   space for fresh job slot
   ($job_{arrival}$, ss),
   job_{exit}=maxU(job_{exit}, D),L^M(job_{exit}, S)>0
4. repeat
5. $D_{old} \leftarrow D$
6. job_{exit}= max U(job_{exit}, D), L(job, S)>0
7. L^M(job_{exit}, S)$\leftarrow$ L^M(job_{exit}, S)-1
8. $job_{arrival}$ = min U($job_{arrival}$, D),
   space for fresh job slot ($job_{arrival}$, ss)
9. until U(job_{exit},D) < U($job_{arrival}$,D_{old})
10. $D \leftarrow D_{old}$
11. end for
12. end for

The scheduler operates in cycles of period C. By deploying a control cycle permits the system to respond rapidly to fresh job submissions and modifications in the length of the tasks noticed for running jobs. In every cycle, the algorithm checks the allocation of tasks on Task-Trackers and their resource utilization. The Task Scheduler governs and manages the allocation decisions, and performance of the system according to the control cycle. The Task Scheduler allocates tasks depending upon the algorithm. As soon as a task finishes, the Task Scheduler chooses fresh task to execute in the available slot, by giving a task of the suitable job to the given TaskTracker.

IV EVALUATION

We present results from evaluation that discover goal of our proposed scheduler which mainly concentrates on enhancing resource allocation. The focus is on resource allocation parameter and compare our scheduler with resource-aware of Hadoop scheduler. In order to evaluate the performance on our scheduler which enhances resource utilization, an accomplishing time goal with each job is created.To evaluate, we included the Grid mix benchmark, which comes along with Hadoop distribution. This includes combine, sort and select. We execute our experiment on a Hadoop cluster encompassing of 2-way 64-bit 2.8GHz Intel Xeon machines. Each machine has 4GB of RAM and runs a 2.6.17 Linux kernel and connected by using Gigabit Ethernet network. The combination with each job instance accomplishing the time goal. The execution of workload on both Fair scheduler and proposed scheduler and compared the results.

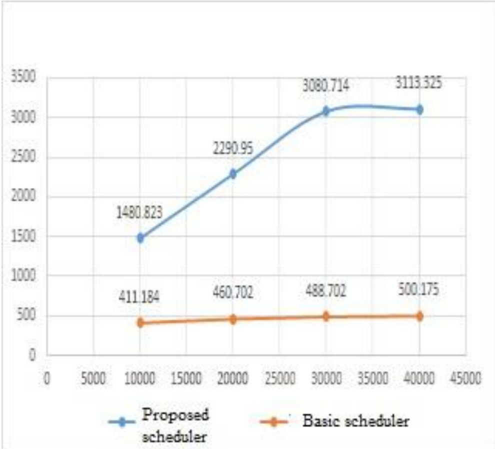

Figure 3: Task scheduling results from schedulers.

The number of map slots per TaskTracker ranges from 1 to 8. Results are shown in Figure 3. The finest static configuration employs 4 concurrent map tasks per TaskTracker. The scheduler surpasses the basic Scheduler for all configurations, presenting an enhancement between 6% and 100%.

V CONCLUSION AND FUTURE WORK

By following the proposed technique, small jobs which take much time to complete will have fair executing performance. In MapReduce framework of distributed computing, every user can divide the identical cluster for myriad of objectives and purposes. As a result various kinds of workloads will execute on the same data centre such as a cluster may be employed for data mining from logs and processing web text which hinge on CPU capability, utilization and I/O bandwidth. The objective of the proposed scheduler algorithm is to find the optimal allocation of tasks as a result to enhance utilization of resource.

REFERENCES

1. J. Dean and S. Ghemawat, “MapReduce: Simplied data processing on large clusters," in OSDI'04, San Francisco, CA, December 2004, pp. 137{150}.
2. Udendhran R, “A Hybrid Approach to Enhance Data Security in Cloud Storage”, ICC '17 Proceedings of the Second International Conference on Internet of things and Cloud Computing at Cambridge University, United Kingdom — March 22 - 23, 2017, ISBN:978-1-4503-4774-7 https://dl.acm.org/citation.cfm?doid=3018896.3025138.
3. Suresh, A., Udendhran, R., Balamurgan, M. et al. "A Novel Internet of Things Framework Integrated with Real Time Monitoring for Intelligent Healthcare Environment "Springer-Journal of Medical System (2019) 43: 165. https://doi.org/10.1007/s10916-019-1302-9.

4. Suresh, A., Udendhran, R. & Balamurgan, M. " Hybridized neural network and decision tree based classifier for prognostic decision making in breast cancers " Springer - Journal of Soft Computing (2019). https://doi.org/10.1007/s00500-019-04066-4.
5. G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, Y. Lu, B. Saha, and E. Harris, "Reining in the outliers in map-reduce clusters using mantri," in OSDI'10. Berkeley, CA, USA: USENIX Asoc., 2010, pp. 1{16}.
6. M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica. Delay scheduling: a simple technique for achieving localityand fairness in cluster scheduling. In Proceedings of the 5th European Conference on Computer systems (EuroSys), 2010.
7. S. Parsa, R. Entezari-Maleki, RASA: A new task scheduling algorithm in grid environment, World Applied sciences journal.
8. K. Liu, H. Jin, J. Chen, X. Liu, D. Yuan, Y. Yang, A Compromised-Time-Cost Scheduling Algorithm in SwinDeWC for Instance-Intensive Cost-Constrained Workflows on a Cloud Computing Platform, Int. J. High Perform. Comput.Appl.
9. K. Kaur, A. Chhabra, G. Singh, Heuristics based genetic algorithm for scheduling static tasks in homogeneous parallel system, Int. J. of Comp. Sci. and Sec.
10. J. Kolodziej, F. Xhafa, Enhancing the genetic-based scheduling in computational grids by a structured hierarchical.
11. R.Rengasamy and M.Chidambaram, "A Novel Predictive Resource Allocation Framework for Cloud Computing", In Proceedings of International Conference on Advanced Computing and Communications Systems (ICACCS), 2019.
12. R.Rengasamy and M.Chidambaram "Challenges and Oppurtunities of Resource Allocation Frameworks for Big data Tools in Cloud Computing", International Journal of Computer Sciences and Engineering, Vol.6, Issue 12. Dec-2018, e-ISSN-2347-2693.

AUTHOR PROFILE

R.Rengasamy is working as Assistant Professor in Bharathidasan Government College for Women (Autonomous), Puducherry. He has completed his PG and M.Phil Degree from Bharathidasan University, Tiruchirapalli, Tamilnadu. He is pursuing Ph.D and his experience in teaching is 15 years. His Area of interest is Cloud Computing and Big Data.

Dr.M.Chidambaram is working as Assistant Professor in Rajah Serfoji Government College (Autonomous), **Thanjavur**. He got his Ph.D from Vinayaga Mission University, Salem, Tamilnadu. He has published more than 50 research papers in reputed national and international journals. His Area of interest includes Grid Computing, Cloud Computing and Big Data. His experience in teaching is 22 years and his experience in research is 5 years.