Agerl Based Enhanced Map Reduce Technique in Cloud Scheduling

S.Selvi[#1], Dr.B.Kalaavathi[#2]

[#1]*Assitant Professor/CSE, Angel College of Engineering and Technology, (Affiliated to Anna University) ,Tirupur, Tamil Nadu, India*

[#2]*Professor/CSE, K.S.R. Institute for Engineering and Technology, Affiliated to Anna University), Tiruchengode, Tamil Nadu, India*

ABSTRACT *- Today's real time big data applications mostly rely on map-reduce (M-R) framework of Hadoop File System (HDFS). Hadoop makes the complexity of such applications in a simpler manner. This paper works on two goals: maximizing resource utilization and reducing the overall job completion time. Based on the goals proposed, we have developed Agent Centric Enhanced Reinforcement Learning Algorithm (AGERL) .The algorithm concentrates in four dimensions: variable partitioning of tasks, calculation of progress ratio of processing tasks including delays, XMPP based multi attribute query posting and Hopkins statistics assessment based dynamic cluster restructuring . An Enhanced Reinforcement Learning Process with the above features is employed to achieve the proposed goal. Finally performance gain is theoretically proved.*

Keywords: *map reduce, Hopkins, multi attribute query, reinforcement learning*

1. INTRODUCTION:

The Map Reduce implementation in a Hadoop framework is an upcoming platform to handle larger applications involving parallel data processing. Research and scientific calculations for a single task can be carried out with substantial parallelism techniques. Improving job completion time and efficient utilization of resources are the main considerations in this framework. This can be achieved when allocation of map and reduce slots are scheduled optimally. The scheduling adopts two types of reinforcement methodologies, one is a discrete process and the other is a continuous process. A continuous reinforcement with a dynamic restructuring mechanism is applied in our work to reschedule the map & reduce tasks whenever there is a necessity occurs.

Identifying and improving the straggler tasks in a heterogeneous cluster[13] is one of the milestones in the process of optimization of scheduling. Our paper enhances the characteristics of RL process. The RL algorithm is composed of the following components: Agent, Environment, State Space and Reward. The characteristics of agent are improved in the existing MRRL scheduler [12] and it performs the following:

a) Variable partitioning of the incoming tasks
b) Computing Progress ratio of the tasks in execution
c) XMPP based multi attribute query model
d) Cluster restructuring based on Hopkins statistical calculation

The agent plays an important role in the identification of straggler tasks and who invokes the multi attribute query based technique to retrieve the requirement of resources needed to convert the straggler tasks into faster tasks.

The environment is the configuration settings of cloud based device. The state representation is the classification of straggler tasks and quicker tasks. The nodes also classified into slow nodes or non scalable nodes and faster nodes or scalable nodes. An objective of RL algorithm is taking actions based on the computation value of reward function. It is positive(>0) and negative(<0) rewards. If the straggler task is able to be executed in the scalable node, then it is a positive reward else it is a negative reward. The scalable nodes can posses more number of staggler tasks with positive reward and non scalable node possess more number of straggler tasks with negative reward.

Previous works [11-13] has not involved the data transfer delays, bandwidth delays in calculating the status of tasks, but in our paper we have included the above delays also for calculation, so that the percentage of accuracy and efficiency can be still improved. Scaling of the resources is made dynamically as like earliest works, but multiattribute based query strategy is adopted in the agent phase of Reinforcement Learning (RL) process to identify the amount of resources left idle at the particular instant. Based on the progress ratio and query results, the agent checks the possibility of the conversion of straggler tasks (including the tolerable percentage of 30%) into quicker one. NOSql is employed for the query retrieval process for transparency.

Section 2 discusses the previous works, section 3 describes the proposed model, section 4 outlines the conclusion and future works.

2. PREVIOUS WORKS:

Implementation of Hadoop Map Reduce Framework is carried out in earlier works and each work was aimed to optimize the framework in various issues like data locality, resource utilization, makespan, performance enhancement,QoS,etc.

Lena et. al. in paper[1] proposed two Energy Aware Map Reduce Scheduling Algorithms EMRSA –I and II to minimize the consumption of energy during the execution of a application, thereby decreasing the overall completion time. Various scheduling algorithms have been studied and analyzed in paper[2]. Liya et. al. worked out the comparision of the mapreduce algorithms by exploring the merits and demerits. Zhuo Tang in paper[3] explored an idea of optimizing the scheduling in a heterogeneous cluster in two phases(job prioritizing phase and task assignment phase).The authors gave a new dimension of scheduling the jobs based on the category of I/O intensive and Compute intensive. Then the jobs are allotted to the machines based on data locality. The paper[4] focuses on a flexible scheduling scheme adopted for any variety of metrics such as deadline, makespan, SLA,etc. Also FLEX aimed to guarantee minimum number of slots to each job and make sure a degree of fairness among the jobs.

A novel scheduler named Resource Aware Adaptive Scheduler (RAS)[5] had been developed to improve the resource utilization and job performance, but RAS cannot resolve the network bottlenecks that occur due to inefficient placement of reduce tasks. Matei et al. in his paper[6] focused on multiuser workloads by adopting two simple techniques namely, delay scheduling and copy compute splitting. The authors applied the concept of statistical multiplexing and achieve high throughput and low response time.

A combination of K-means clustering and self adaptive map reduce had been employed to improve the performance of scheduling by doing less computation and providing higher accuracy[7]. But Thangaselvi et al. proposed her work by assigning only one task for each data node which failed to improve the overall throughput of multiple task.

An iterative computation was done by Ekanayake et al. in [8] and the authors proposed a programming model and architecture in a map reduce framework and showed the improvement in the execution time of incoming jobs. Based on input data, machine slots and complexity of reduce function, a cost function was proposed by Tian et al. in paper[9] and showed the minimization of cost within the job deadline and budget.

Paper[10] focused on demand resource provisioning by employing a two tiered allocation mechanism ie, locally and globally and a feedback mechanism was also employed to manage the on demand capacities to the concurrent applications.

Pastorelli et al. in his paper[11] evaluated job size online in a multi server system using aging technique and achieved fault tolerance, scale-out upgrades and attempt to eliminate the starvation issues both for smaller and larger jobs.

Reinforcement Learning technique [12] is adopted in a heterogeneous cluster to identify and reduce the straggler tasks .Nenavath et al. tried to improve the utilization of resources and minimizes the overall job completion time. But we have extended the RL by assessing using Hopkins statistics to guide in cluster restructuring.

Dynamic Proportional Share Model is designed using multiattribute range query technique in paper[13] and the query model is extended by adopting XMPP protocol in our paper.

3. THE PROPOSED AGERL ALGORITHM:

3.1. HOPKINS STATISTICS ASSESSMENT BASED CLUSTER RESTRUCTERING:

Map Reduce Framework works in a heterogeneous cluster and when the user submits the tasks, the task is partitioned into required number of map and reduces tasks in a non homogenous manner. The partition takes place based on the configuration of machines in the cluster, nature of tasks and data locality. For this, the clustering tendency is assessed based on Hopkins statistics methodology.

3.1.1. Procedure to perform Hopkins Statistics Assessment:

Step 1: Assume the set of points within a node $n_k = \{p_1,p_2,\ldots,p_n\}$

Step 2: For every point p_i in the node n_i and its nearest neighbouring point p_j , compute the Manhautten Distance between p_i and p_j and it is represented as

U_m = Manhautten Distance($p_{i,}$ p_j).

Manhautten Distance($p_{i,}$ p_j) = |x1 – x2|+|y1-y2|,

where (x1,y1) and (x2,y2) are the coordinates of p_i and p_j respectively.

Step 3: For every points p_i and p_j within the node $n_{k,}$ compute the Manhautten Distance between p_i and p_j and it is represented as

V_m = Manhautten Distance(p_i,p_j).

Step 4: Compute Hopkins Statistics (H)

$$H = \frac{\sum U_m}{\sum U_m + \sum V_m}$$

Step 5: H value for every node is computed in the similar manner and if H(avg) > 0.5,then the cluster to be restructured would be a meaningful cluster.

3.2. PROGRESS RATIO CALCULATION:

Assume the task is splitted into m number of map tasks $M=\{M_1,M_2,\ldots,M_m\}$ and n number of Reduce tasks $R=\{R_1,R_2,\ldots,R_n\}$

The current status of the tasks in execution is calculated on a periodic basis and is given by

Progress Ratio of Task (M_i/R_i) =

$$\left(\frac{\text{Length of the task} - \text{Amount of Task Completed}}{\text{Length of the task}} + (DT_i + BD_i) \right) \times 100 \qquad (1)$$

$$\text{Progress Ratio avg} = \frac{\sum_{i=1\text{-}m} (M_i) + \sum j_{=1\text{-}n} (R_i)}{(m+n)} \qquad (2)$$

Nenavath et al. [12] set the tolerable limit to 20% and we have considered data transfer delay (DT_i) and bandwidth delay(BD_i) to improve the accuracy of scheduling.

3.3.XMPP BASED MULTI ATTRIBUTE QUERY MODEL:

It is the responsibility of the agent to identify suitable node for the processing of straggler tasks. The agent based on the progress ratio of each tasks posts query to the nodes in the cluster. Extensible Messaging and Presence Protocol (XMPP) based multiattribute query model is employed in our work.

XMPP protocol works as follows: A query message is send to the machine A. If the message is send to the right node, then the execution proceeds, otherwise A itself sends it to the next nearest optimal node. This continues until the right node is found.

The query is first posted to the nodes with data locality and based on the reply (if that node's resources are able to process the straggler task in a faster mode), restructuring of the cluster is predicted with Hopkins statistical assessment. If the assessment results above 0.5, then the prediction are preceded, otherwise the query is posted to nearest nodes and the same procedure is repeated till the prediction becomes correct. The query structure is evaluated in NoSQL. The query includes the attributes such as CPU, memory, Data Transfer and Bandwidth. These resources are included as a part of the query. The result of the query posted to a node gives the information whether that node can run the straggler task, otherwise restructuring is predicted with Hopkins. If not, the query is posted to the nearest neighbor nodes(secondary priority nodes). This process goes on until a suitable node is selected, if not that conversion of that particular straggler task into a quicker one has to be dropped. In this manner, we try to reduce the number of straggler tasks in the heterogeneous cluster.

3.4. PSEUDOCODE : PROPOSED AGERL ALGORITHM

Step 1: Get the task T as input from the user

Step 2: Job Tracker analyze the task for its characteristics

Step 3: From the characteristics, Agent A performs variable partitioning of the tasks into m Map tasks and n Reduce tasks

Step 4: for each Mi/Ri tasks

compute the $PR(M_i/R_j)$ & PR_{avg} (as in Eqn 1 & 2)

Shoot a query to the higher priority nodes

Compute the reward.

if reward>0

possible to convert the straggler tasks into a quicker task

compute Hopkins statistics(H) and check

if (H)>0.5

restructure the cluster

Step 5: Compute performance gain.

4. CONCLUSION:

The paper proposed an Enhanced version of Reinforcement Learning Approach in the aim of optimizing the scheduling in Map Reduce Framework. The agent centric RL approach is extended in four dimensions: partitioning the task into map and reduce tasks based on the variable nature of nodes, minimizing the number of straggler tasks based on progress ratio and predicting to convert into a quicker one, XMPP based multi attribute query posting to identify the right node to process the straggler tasks, restructuring the cluster dynamically based on Hopkins statistical assessment. The agent in RL process plays an important role for minimizing the overall job completion time and effective resource utilization. We have tried to improve the accuracy of optimization by setting the tolerable limit to 30% and including the delays for progress ratio calculation. As a part of future work, other methodologies can be incorporated for variable partitioning and cluster restructuring to further improve the effectiveness in scheduling.

5. REFERENCES:

[1] Lena Mashayekhy, Mahyar Movahed Nejad, Daniel Grosu, Quan Zhang, Weisong Shi, "Energy-Aware Scheduling of MapReduce Jobs for Big Data Applications", IEEE Transactions on Parallel and Distributed Systems, vol. 25,September 2014.

[2] Liya Thomas, Syama R," Survey on MapReduce Scheduling Algorithms", International Journal of Computer Applications (0975 – 8887) Volume 95– No.23, June 2014.

[3] Zhuo Tang, Min Liu, Kenli Li, Keqin Li," An Optimized MapReduce Workflow Scheduling Algorithm for Heterogeneous Computing", The Journal of Supercomputing, Volume 72 Issue 6, June 2016,Pg no. 2059-2079.

[4] Joel Wolf, Deepak Rajan, Kirsten Hildrum, Rohit Khandekar, Vibhore Kumar, Sujay Parekh, Kun-Lung Wu and Andrey Balmin," FLEX: A Slot Allocation Scheduling Optimizer for MapReduce Workloads", ," in Proc. ACM/IFIP/USENIX 11th Int. Conf. Middleware, 2010, pp. 1–20.

[5] Jord`a Polo, Claris Castillo, David Carrera, Yolanda Becerra, Ian Whalley, Malgorzata Steinder2, Jordi Torres, and Eduard Ayguade," Resource-Aware Adaptive Scheduling for MapReduce Clusters", Proceedings of 12th International Middleware Conference, Lisbon, Portugal, pg no. 187-207 ,December 12-16, 2011.

[6] Matei Zaharia, Dhruba Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy,Scott Shenker,Ion Stoica," Job Scheduling for Multi-User MapReduce Clusters", EECS Department University of California, Berkeley Technical Report No. UCB/EECS-2009-55,April 30,2009.

[7] R.Thangaselvi, S.Ananthbabu, R.Aruna," An Efficient Mapreduce Scheduling Algorithm in Hadoop", International Journal of Engineering Research & Science (IJOER), Vol-1, Issue-9, December, 2015.

[8] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox, "Twister: A runtime for iterative MapReduce," in Proc. 19th ACM Int. Symp. High Performance Distributed Computing, pg no.810–818,2010.

[9] A. Verma, L. Cherkasova, and R. H. Campbell, "ARIA: Automatic resource inference and allocation for MapReduce environments,"in Proc. 8th ACM Int. Conf. Autonomic Comput., 2011, pp. 235–244.

[10] Y. Song, Y. Sun, and W. Shi, "A two-tiered on-demand resource allocation mechanism for VM-based data centers," IEEE Trans. Services Comput., vol. 6, no. 1, pp. 116–129, Jan. 2013.

[11] M. Pastorelli, A. Barbuzzi, D. Carra, M. Dell'Amico, and P. Michiardi,"HFSP: Size-based scheduling for Hadoop," in Proc. IEEE Int. Conf. Big Data, 2013, pp. 51–59.

[12] Nenavath Srinivas Naik, Atul Negi, V. N. Sastry, "Performance Improvement of MapReduce Framework in Heterogeneous Context using Reinforcement Learning", 2nd International Symposium on Big Data and Cloud Computing,2015.

[13]Gayathri, Selvi and Kalavathi " An Efficient Performance and Monetary Cost Optimization on Resource Allocation in Cloud" International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 5, May 2015.

Authors Bibilography

S.Selvi is currently doing Ph. D. degree in the area of Cloud Computing in Anna University,Chennai and working as an Assistant Professor in the department of Computer Science and Engineering at Angel College of Engineering, Tirupur. She received her B.E. degree in 2002 and M.E. degree in 2009. Her area of interests includes Cloud Computing ,Compiler Design and Data Structures.

Dr. B. Kalaavathi received B.E. (Computer Science and Engineering) degree in 1993 from Bharathiyar University, M.Tech from Pondicherry University in 2000 and Ph.D from Periyar University in 2010. She is currently working as a Professor/Head in the Department of CSE in K.S.R. Institute for Engineering andTechnology,Tiruchengode. She has about 19 years of experience in teaching. She is a member of CSI & ISTE, INDIA. Her current areas of interest include Mobile Computing, Data Structures and Algorithms and Wireless Networks

APPENDIX DATA FLOW DIAGRAM OF AGERL ALGORITHM: