Research on MapReduce Task Scheduling Optimization

Ying Ren[1,a,] **Huawei Li**[2,b] **and Lina Wang**[1,c]

[1]Naval aeronautical university, Shandong, China
[2]Shan dong Businss Institute, Shandong,Yantai, 264001, China
Email: [a]ren.ying@163.com; [b]li_huawei@163.com; [c]wanglina5677@163.com

Abstract. One of the big challenges in MapReduce is how to effectively allocate resources for the jobs submitted by users , and finish the job within the time specified by the user .The original scheduling strategy lacks the ability of job management and can not complete the job according to the time requirement of the user. In this paper, a task scheduler (ET-scheduler) is proposed to meet the needs of job time and resource optimization. It can not only meet the time needs of users, but also minimize the resource consumption and adjust the time allocation in the process of map and reduce. Experiments show that the algorithm not only completes the most jobs in a given time, but also minimizes the resource consumption in the Hadoop cluster.

1.Foreword

The task scheduler is responsible for task scheduling and the allocation and management of cluster resources. An efficient task scheduler can effectively improve the execution time of jobs in the cluster. The default scheduler of MapReduce only considers single task data localization. When multiple tasks arrive at the same time, it can not satisfy the simultaneous localization of multiple tasks at the same time, making the execution time of the job significantly increased. In this paper, a task scheduler (called ET-scheduler) is proposed to meet the needs of time and resource optimization. The main goal is to improve the localization of data, meet the user's demand for running time and minimize the consumption of resources. A task scheduling algorithm is proposed to meet the needs of job time and resource optimization. The job description file is established to estimate the completion time of the job and to calculate the slot lotted resources required for the job completion time. The algorithm can not only meet the time needs of the user, It also minimizes resource consumption and reasonably adjusts time allocation in map and reduce processes.

2.Hadoop task scheduling process

The task scheduling flow of Hadoop is as shown in figure 1, and the main task scheduler is the task scheduler. When the user submits a job, the job related task scheduling is managed by Task Scheduler. The nodes in Hadoop cluster can be divided into master node and slave node master node, which can also be called management node. In general, the task scheduler and resource management module are located in the cluster management module, which is called computing node or DataNode node. It is usually responsible for the storage of the tasks and data blocks assigned by the master node and reporting to the master node the usage of slot lots. The task scheduler in Hadoop is usually located in the master node, and the master node is responsible for managing the DataNode node in the cluster. Each DataNode node will allocate a certain slot(MapReduce to manage the cluster resource, including map slot and reduce slot. The DataNode node in the cluster periodically reports to the master node the progress of the remaining slot and the current execution of the task, and master allocates other tasks for them according to the task scheduling rules and the rules allocated by the residual slot. Figure 1 illustrates the detailed flow of task scheduling.

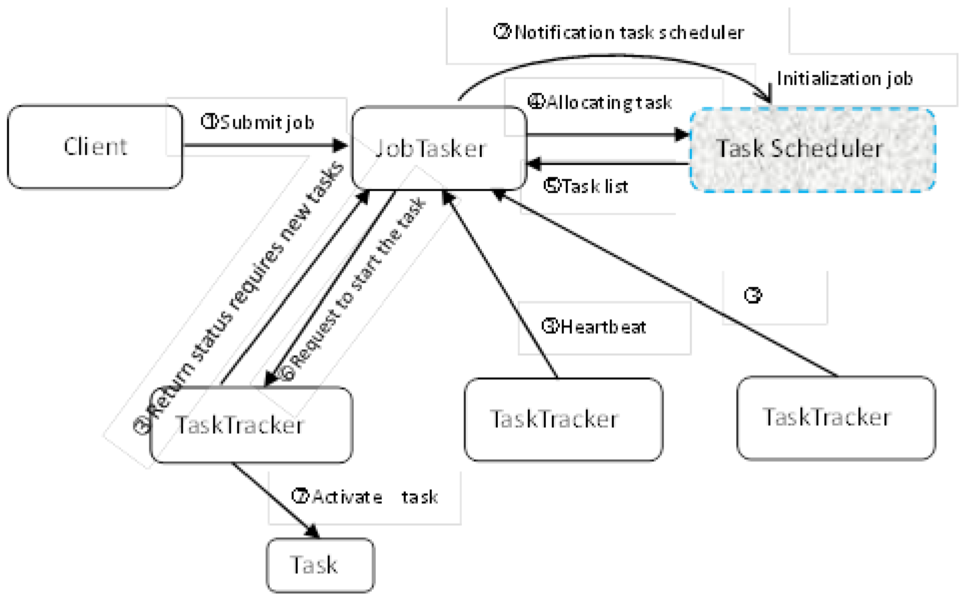

Figure1.The task scheduler process in Hadoop

3. Uncertainty of Job execution time in Task scheduling

A job in a MapReduce needs to be assigned a slot when it is executed. The slot is the unit slot of the MapReduce partition resource and can be divided into map slot and reduce slot. The length of the job execution time related to the number of the map slot and reduce slots which are assigned to the job. It is very difficult to accurately predict the execution time of a task, because even if the same job, if the slot resources assigned to it are different, then the execution time of the job is also very different.

4.Task scheduling algorithm satisfying job time requirements and resource optimization

It is very difficult to accurately predict the execution time of a job, which is affected by resource allocation, node performance and data localization. In recent years, the performance of job execution in MapReduce is restricted by many factors, so it is difficult to establish a complete prediction model, so the method of job description file is being studied extensively.

4.1.Establishment of Job description File

The purpose of establishing the job description file is to describe the time consumption of the job in the maps、shuffle、sort and reduce phases and the resource consumption on each slot in the Hadoop cluster. The job description file created in this section more accurately records the overlap time of the map and shuffle phases and resource consumption. These data information can be obtained through the log files of the jobs in the master node in the Hadoop cluster, or through the cluster monitoring tool Ganglia .

The Map phase is composed of many map tasks, and in order to accurately describe the process, this article defines the following attributes for the Map phase in the job description file:$(\mathrm{Map}_{\mathrm{min}}, \mathrm{Map}_{\mathrm{avg}}, \mathrm{Map}_{\mathrm{max}})$, $\mathrm{Map}_{\mathrm{min}}$is the shortest time of map task execution. Its main function is not to estimate the minimum execution time of Map task, but to estimate the beginning time of shuffle phase when the first Map task ends. $\mathrm{Map}_{\mathrm{avg}}$ is the average time a Map task is executed. $\mathrm{Map}_{\mathrm{max}}$is the maximum time required for Map task execution. Resource consumption is defined as the em_j^J, and its meaning is resource consumption on the jth map slot when the job J is executed.

The Reduce phase takes place after the shuffle phase is completed, and the attributes defined in the reduce phase in the job description file are:$(\mathrm{Reduce}_{\mathrm{avg}}, \mathrm{Reduce}_{\mathrm{max}})$. $\mathrm{Reduce}_{\mathrm{avg}}$ is the average time Reduce task is executed. $\mathrm{Reduce}_{\mathrm{max}}$is the maximum time required for Reduce task execution.

Resource consumption is defined as the er_j^J, its meaning is resource consumption on the j th reduce slot when the job J executes.

4.2.Calculation of job completion time

Although it is difficult to estimate the execution time of a job accurately in the MapReduce model, the upper and lower bounds of the execution time of the job can be estimated. Assume a job is divided into n tasks, and the time each task is executed is represented by a set of $\mathrm{Time} = \{\mathrm{Time}_1, \mathrm{Time}_2, \mathrm{Time}_3, \mathrm{Time}_n\}$.The number of slot in a cluster that can perform tasks at the same time is k. So for these n tasks, the average task execution time is

$$\mathrm{avgTime} = \frac{\sum_{i=1}^{n} Time_i}{n} \tag{1}$$

When a new data set arrives, the job J is divided into $\mathrm{C}_\mathrm{M}^\mathrm{J}$map task and $\mathrm{C}_\mathrm{R}^\mathrm{J}$ruduce task, since the resources that execute the task are partitioned by slot, so the slot used to execute the map task and the reduce task is called map slot and reduce slotted, so the corresponding number of map slot and reduce slot are: $\mathrm{L}_\mathrm{M}^\mathrm{J}$and $\mathrm{L}_\mathrm{R}^\mathrm{J}$.

1)Calculation of map phase time

The average execution time of map task and the maximum execution time are recorded in the description file of job J.In this paper, $\mathrm{Time}_\mathrm{M}^\mathrm{low}$and $\mathrm{Time}_\mathrm{M}^\mathrm{up}$ are used to represent the minimum execution time and maximum execution time of job J in the map phase, respectively. The formula for the minimum execution time and maximum execution time of operation J in the map stage is:

$$Time_M^{low} = \frac{C_M^J \cdot Map_{avg}}{L_M^J} \tag{2}$$

$$Time_M^{up} = \frac{(C_M^J - 1) \cdot Map_{avg}}{L_M^J} + Map_{max} \tag{3}$$

2)Time calculation of reduce stage

In this paper, $\mathrm{Time}_\mathrm{R}^\mathrm{low}$ and $\mathrm{Time}_\mathrm{R}^\mathrm{up}$ are used to represent the minimum execution time and maximum execution time of job J in the reduce phase, respectively.The formulas for the minimum execution time and maximum execution time of job J in the reduce phase are as follows:

$$Time_R^{low} = \frac{C_R^J \cdot Reduce_{avg}}{L_R^J} \tag{4}$$

$$Time_R^{up} = \frac{(C_R^J - 1) \cdot Reduce_{avg}}{L_R^J} + Reduce_{max} \tag{5}$$

3)Time calculation for shuffle and sorting phases

$\mathrm{Time}_\mathrm{shuffle}^\mathrm{low}$ and $\mathrm{Time}_\mathrm{shuffle}^\mathrm{up}$ respectively indicate the minimum execution time and the maximum execution time of the job J in the normal shuffle stage.The formulas for the minimum execution time and maximum execution time of job J in the shuffle phase can be obtained as follows:

$$\mathrm{Time}_\mathrm{shuffle}^\mathrm{low} = \left[\frac{\mathrm{C}_\mathrm{R}^\mathrm{J}}{\mathrm{L}_\mathrm{R}^\mathrm{J}} - 1\right] \cdot \mathrm{Shuffle}_\mathrm{avg} \tag{6}$$

$$Time_{shuffle}^{up} = \left[\frac{C_R^J}{L_R^J} - 1\right] \cdot Shuffle_{avg} + Shuffle_{max} \tag{7}$$

$\mathrm{Time}_{\mathrm{shuffle\_one}}^{\mathrm{low}}$ and $\mathrm{Time}_{\mathrm{shuffle\_one}}^{\mathrm{up}}$ respectively indicate the minimum execution time and maximum execution time of job J in the first shuffle stage. In calculation $\mathrm{Time}_{\mathrm{shuffle\_one}}^{\mathrm{low}}$, the time when map overlaps with the first shuffle is subtracted.The formula for the minimum execution time and maximum execution time of the job J in the shuffle phase is :

$$Time_{shuffle\_one}^{low} = \frac{C_R^J}{L_R^J} \cdot (Shuffle1_{avg} - Shuffle1_{overlap\_avg}) \tag{8}$$

$$Time_{shuffle\_one}^{up} = \left[\frac{C_R^J}{L_R^J} - 1\right] \cdot Shuffle1_{avg} + Shuffle1_{max} \tag{9}$$

Thus the formulas for the minimum execution time $\mathrm{Time}_{\mathrm{J}}^{\mathrm{low}}$ and maximum execution time $\mathrm{Time}_{\mathrm{J}}^{\mathrm{up}}$ of job J are as follows:

$$Time_J^{low} = Time_M^{low} + Time_{shuffle\_one}^{low} + Time_{shuffle}^{low} + Time_R^{low} \tag{10}$$

$$Time_J^{up} = Time_M^{up} + Time_{shuffle\_one}^{up} + Time_{shuffle}^{up} + Time_R^{up} \tag{11}$$

4.3.Calculation of slot required for work
Job J assigns appropriate slot to the job under the condition of the given execution time, input data and job description file to meet the needs of user time. Job J is required to be completed within a given time and the slot used is minimal. Map slot and Reduce slot are represented by$\mathrm{L}_{\mathrm{M}}^{\mathrm{J}}$and $\mathrm{L}_{\mathrm{R}}^{\mathrm{J}}$, so that slot is minimized. In the process of job J execution, controlling the proportion of map task and reduce task execution time can effectively prevent the reasonable time allocation of map and reduce task, which can effectively increase the task execution time. Although the relationship between execution time and slot can be established, however, the rational allocation of map task execution time and reduce time is ignored. In order to allocate the execution time of map task and reduce task more reasonably, this paper adds the constraint condition to control the execution time of map and reduce task.

5.Experiment and analysis

The specific experimental scheme is to input data of different sizes in the test program sort、 word count and k-means, and select n different jobs at random, and produce a job queue of n size, which can be represented by set Job={Job_1,Job_2 〖,Job〗 _3,…,Job_n}.Each Job consists of three elements, Job_i=(i,Tdd_i,D_i).i is the sequence number of the job and Tdd_iis the time of arrival of the job and D_i is the expected execution time of the job, that is, the specified time (in seconds),and submit the job queue to Hadoop for testing. If(1,11.23.14,1000)indicates that job 1 wants to complete in 1000 seconds, the arrival time is 11: 23: 14.The completion time for each job is marked as Twc_i.W represents a set of the number of jobs that have not been completed in a given time. Therefore, in order to evaluate the effectiveness of the scheduler proposed in this paper, the following evaluation indicators are introduced:
1)Out of time ratio

$$\sum_{Job\in W} \frac{Twc_i - Tdd_i - D_i}{D_i} \tag{12}$$

The index counts all the proportions that exceed the specified time, and all the time units in the formula are seconds.
2)percentage of operations not completed within the required time

$$Count(W)/_n \tag{13}$$

3)Number of operations completed:$n - Count(F)$.
100 jobs are inserted randomly into the Job queue, and the jobs are interpolating in a certain time interval. Time intervals are randomly generated, but instead of try not to insert more jobs at the same time, they are inserted at certain intervals, because different clusters have different numbers of compute nodes, so the processing power of jobs varies. Within a certain period of time, there is a limit on the maximum number of assignments. The 100 test jobs that insert the job queue are randomly selected in three algorithms, Sort, WordCount, and K-means, that is, the 100 test jobs are composed of three algorithms Sort, WordCount, and k-means. The input size of the data is controlled below 1GB. The job description file for the program under test is assumed to have been generated. The implementation results are shown in figures 2 and 3.

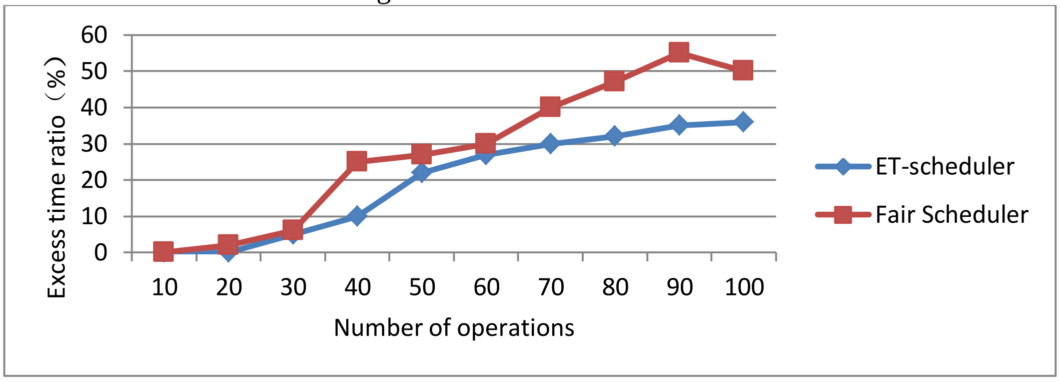

Figure.2 The influence of the number of jobs for deadline exceeded

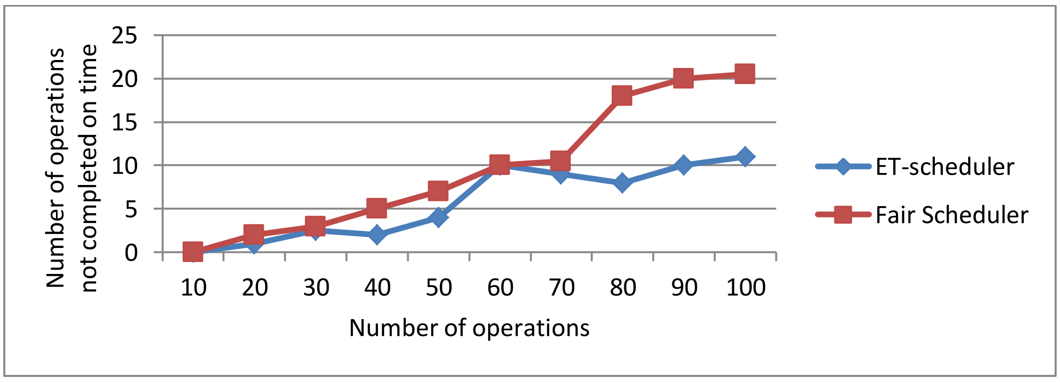

Figure.3 The influence of the number of jobs for missed-deadline

Figure 2 and Figure 3 illustrate the effect of the number of jobs on the performance of the scheduler. It can be seen that the number of jobs not completed by the ET-scheduler within the specified time is relatively small, and the ratio of time exceeding the specified time of the job is also the smallest. It can be seen from figure 2 that the ET-scheduler scheduler shows relatively stable performance when the number of jobs is greater than 60.This is because when the size of the Hadoop cluster is fixed, its corresponding Map slot and reduce slot are also relatively fixed, and only a limited number of assignments can be completed within the specified time. It can be seen that when the number of assignments is less than or equal to 40, the number of tasks that are not completed on time is lower, and when the number of assignments is less than 10, all submitted tasks can be completed on time. However, with the increase of the number of jobs, the number of jobs that are not completed by time also increases obviously with the increase of the number of Fair Scheduler schedulers. Compared with the historical data of Fair Scheduler scheduler, the average job execution time and the minimum execution time are estimated accurately.

In addition, the scheduler dynamically recalculates the slots required by the job during execution, which can release more slot for use by other jobs, especially small jobs that require less slot. The Fair Scheduler scheduler that comes with Hadoop only assigns resources fairly to each task as far as possible, and does not specifically optimize the slot required by the job and the time requirements of the job. Therefore, as the number of jobs increases, the out-of-time ratio of the Fair Scheduler scheduler and the tasks that are not completed as required are more. The ET-scheduler scheduler proposed in this paper calculates the minimum time required for job completion according to the job description file. The job description file is generated according to the operation of the job on different data sets, and the calculation error of the time is relatively small. The dynamic adaptive replica placement algorithm proposed in this paper can dynamically increase the number of replicas of hot data blocks, and the validity of data localization is superior to that of previous algorithms.

6.Section of this chapter

This paper first analyzes the shortcomings of the existing task scheduling policies in MapReduce, then analyzes the uncertainty of the job execution time, and points out that the job execution time is directly related to the slot resources allocated by the job. The job completion time and slot consumption are estimated by establishing the job description file and according to the time and resource consumption information in the job description file. Finally, a task scheduling algorithm is proposed to meet job time requirements and resource optimization. The algorithm can not only satisfy the time requirements of users, but also minimize resource consumption. The experimental results show that the task scheduling algorithm designed in this paper not only completes the most jobs in the specified time, but also minimizes the resource consumption in the Hadoop cluster.

7.References

[1] Taha Osman,Dhawal Thakker,David A1-Dabass.Semantic-driven matchmaking of web services using case-based reasoning[C]//Web Services,206.ICW0S'06.International Conference on.IEEE,2006:29-36.
[1] Tang Z, Zhou J Q, Li K L, Li R X. A MapReduce task scheduling algorithm for deadline constraints [J]. Cluster Computing, 2013, 16(4):651-662.
[2] Negi A, Sastry V N. A Review of Adaptive Approaches to MapReduce Scheduling in Heterogeneous Environments [C]. Proceeding of 3rd International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, India, 2014, 677-683.
[3] Verma A, Cherkasova L, Automatic resource inference and allocation for mapreduce environments [C]. Proceedings the the 8th ACM International Conference on Autonomic Computing, 2011, 235-244.
[4] Dubois E, Falcao J, Leonard M. Towards an executive master degree for the new job profile of a service systems innovation architect [C]. Proceedings of 2011 Annual SRII Global Conference, 2011, 572-578.
[5] Mar á L L G, Ricardo G, Antonia G G. K-means algorithms for functional data [J].Neurocomputing, 2015, 151:231-245.
[6] Gu C, Chen Q, Tao Q. An improved K-means algorithm combined with chaotic particle swarm optimization algorithm [J]. Journal of Information and Computational Science,2015,12(10):4113-4124.
[7] Liu G, Li S, Weng B.K-means clustering based on interactive differential evolution for color image clustering [J]. ICIC Express Letters, 2015, 9(8):2327-2334.
[8] Xie K, Wu J, Yang W K, Changyin Sun. K-Means clustering based on density for scene image classification [J]. Lecture Notes in Electrical Engineering, 2015, 336:379-386.
[9] Wu J, Liu H, Cao X, Chen J. K-means-based consensus clustering: A unified view [J].
[10] Xiong R, Luo J, Dong F. Optimizing data placement in heterogeneous Hadoop clusters [J].Cluster Comput, 2015,18:1645-1680.

[11] Hong W, Zheng T. Wang H. Dynamic user profile-based job recommender system[C].Proceedings of the 8th International Conference on Computer Science and Education, 2013,1499-1503.
[12] Heap B, Kizywicki A, Wobcke W, Bain M, Compton P. Combining career progression and profile matching in a job recommender system [J]. Lecture Notes in Computer Science(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, 2014, 8862:396-408.
[13] Wu M, Zhang Z, Li Y. Application research of Hadoop resource monitoring system based on Ganglia and Nagios [C]. Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, 2013,684-688.
[14] Ganapathi A, and Chen Y et al. Statistics-driven workload modeling for the cloud [C]. IEEE 26th International Conference on Data Engineering Workshops, 2010,87-92.
[15] Mingming S, Hang Z, Xuehai Z, Kun L, Changlong L. HPSO: Prefetching Based Scheduling to Improve Data Locality for MapReduce Clusters[C], Lecture Notes in Computer Science(including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2014, 8631:82-95.