

# Scheduling Optimization Algorithm Based on Hadoop

Qingnian Zhang, Ying Gao, Zhao Chen, and Xiuxia Zhang

**Abstract**—With the increasingly sophisticated and popularity of the third-generation communication networks, the size of the network is drastically becomes larger and the equipment complexity is also greatly improved. So how to quickly resolve the call failure and how to ensure the quality of network operation becomes the most urgent task for network maintenance personnel. Call trace system, a new feature derived from signaling trace, as an important subsystem of the network management system is a powerful means to solve call failure problem and guarantee the quality of the network operation. Scheduling in call tracking system used in communication is fifo scheduling algorithm of Hadoop. In practical applications, in order to make full use of the resources, different levels of users sharing a cluster, the different types of call tracking data processing will be done in the same cluster. Therefore, on the basis of the fair scheduling, this paper put forward based on the improved algorithm of call tracking system, to improve the system of the department can, better meet the actual demand of the system.

**Index Terms**—Scheduling algorithm, system optimization, scheduling algorithm, Hadoop.

## I. INTRODUCTION

Job scheduling algorithm is one of the core technology of Hadoop, is now a major direction of domestic and foreign scholars study the Hadoop. Scheduling algorithm is mainly responsible for operation in the system of execution sequence and cluster resources allocation problem, namely to choose the appropriate task allocation to the appropriate TaskTracker execution. The scheduling process is very complicated, good scheduling can significantly improve the overall performance of the cluster, improve the cluster resource utilization; May lead to cluster conversely some TaskTracker under the heavy load, low efficiency, poor user experience problems [1], [2].

## II. BASED ON THE COMBINATION OF THE THRESHOLD TIME PRIORITY SCHEDULING ALGORITHM RESEARCH

### A. Algorithm Improvement Ideas

The first level of system resources allocation is depending on the type of call tracking data, the default secondary resource allocation fair scheduling algorithm is priority in accordance with the homework, so in the process of the allocation of resources, does not take into account the user priority influence on the allocation of resources, so need to improve the operation of weight calculation formula, the user priority considering weight calculation in the job [3]-[5]. For super users to submit homework, only when the job waiting

time reached the second waiting time threshold, the operational priorities will become a super user can submit a job the maximum weight. Set a regular user priority with  $u_{low}$ , said the priority of the super user use  $u_{high}$  said [6], [7].

Based on the analysis of above, the following will be the concrete analysis of the new improved algorithm needs to meet the conditions.

#### 1) The user priority and task priority handling

Assumes the user priority is with variable  $u$ , assignment priority is with variable  $j$  said that if the user submitted  $u1, j_{11}, j_{12}, j_{13}, j_{14}, \dots, j_{1N}$ ,  $N$  is homework, user submitted  $u2, j_{21}, j_{22}, j_{23}, j_{24}, \dots, j_{2M}$ ,  $M$  is homework, it first determines the user priority, the process is as follows:

- 1) If the  $u1 > u2, j_{11}, j_{12}, j_{13}, j_{14}, \dots, j_{1N}$ , weight value should be not less than the weight of  $j_{21}, j_{22}, j_{23}, j_{24}, \dots, j_{2M}$ . According to the operation priority and user priority combination method to calculate weight assignment, and queued, from big to small order scheduling execution in turn.
- 2) If  $u1 = u2, j_{11}, j_{12}, j_{13}, j_{14}, \dots, j_{1N}$  and  $j_{21}, j_{22}, j_{23}, j_{24}, \dots, j_{2M}$  weight should be the same does not consider the size of the relationship between the user to the same priority, then according to the operation priority and user priority combination method to calculate weight assignment, and queued, from big to small order scheduling execution in turn [8].

#### 2) Based on time weight updating threshold homework

A new configuration file, in the configuration file, set properties Time Threshold, said wait time threshold, which weights updated boundary value. Homework to weight *job weight* satisfy the following conditions:

- 1) When *Wait Time* < *Time Threshold*, weight meet formula (1)

$$\text{jobWeight} = \begin{cases} \text{jobWeight}(\text{tasksNum}, u, j), \\ \text{considering job size} \\ \text{jobWeight}(u, j), \text{ othersize} \end{cases} \quad (1)$$

Namely when the job waiting time does not exceed the time threshold, the weight of homework is has nothing to do with the time.

- 2) When *Wait Time*  $\geq$  *Time Threshold*, both ordinary users and super users to submit homework, satisfy all the operation weight formula (2).

$$\text{jobWeight} = \begin{cases} \text{JobWeight}(\text{tasksNum}, u_{high}, \max(j)), \\ \text{considering job size} \\ \text{JobWeight}(u_{high}, \max(j)), \text{ othersize} \end{cases} \quad (2)$$

#### 3) Homework share calculation

Operation weight value is closely related to the operation of

resources. Each job have corresponding fair share resources for their execution *Fair Share*, theoretically the more the greater the weight value of resources. In Hadoop, how much resources is by the number of slot tasks, including Map and Reduce tasks are calculated separately. Assuming that the current job pool available task slot for *Pool Map Capacity* Map phase, then the pool one job fair share resources *Fair Share* calculation as shown in formula (3).

$$\begin{aligned} \text{Fair Share}_i &= \frac{\text{jobWeight}_i}{\sum_{i=1}^M \text{jobWeight}_i} \times \text{pool map Capacity} \\ &= \frac{j_i}{\sum_{i=1}^M j_i} \times \text{pool map Capacity} \end{aligned} \quad (3)$$

Among them, the  $M$  pool the total number of resources for the assignments. Reduce phase how resources are to be divided with the Map phase [9]-[11].

### B. Algorithm Design and Implementation

#### 1) Based on user priority and task priority weighting combination of homework

Assuming *job Weight* = *job Weight* ( $u, j$ ). Are under the same user priority, different priority assignment should get the resources with the original does not consider user priority without too big change, simplified as shown in Eq. (4).

$$\begin{aligned} \frac{j_i}{\sum_{i=1}^M j_i} &= \frac{\text{jobWeight}(u_i, j_i)}{\sum_{i=1}^M \text{jobWeight}(u_i, j_i)}, \\ \text{when } u_i &= u_j, 1 \leq i, j \leq M \end{aligned} \quad (4)$$

Because all the  $u_i$  values are the same, equation (4) can be simplified, the result as shown in equation (5).

$$\frac{j_i}{\sum_{i=1}^M j_i} = \frac{\text{jobWeight}(u, j_i)}{\sum_{i=1}^M \text{jobWeight}(u, j_i)} \quad (5)$$

The  $u$  is for fixed value,  $1 \leq i, j \leq M$ .

According to the mathematical knowledge, function  $y = ax, a > 0$  can accord with the requirement of job Weight ( $u, j_i$ ), as shown in equation (6).

$$\text{jobWeight}(u, j) = u \times j \quad (6)$$

Customer priority is not same, is a high-priority job to obtain more resources than the original resource assignments. Existing in the system user priority  $u_{\text{low}}$  and  $u_{\text{high}}$ , should satisfy the equation (7).

$$\begin{cases} \frac{j_k}{\sum_{k=1}^N j_k} \geq \frac{\text{jobWeight}(u_{\text{low}}, j_k)}{\sum_{k=1}^N \text{jobWeight}(u_{\text{low}}, j_k) + \sum_{l=1}^{M-N} \text{jobWeight}(u_{\text{high}}, j_l)} \\ \frac{j_l}{\sum_{l=1}^M j_l} \leq \frac{\text{jobWeight}(u_{\text{high}}, j_l)}{\sum_{k=1}^N \text{jobWeight}(u_{\text{low}}, j_k) + \sum_{l=1}^{M-N} \text{jobWeight}(u_{\text{high}}, j_l)} \end{cases} \quad (7)$$

where  $N$  represents the homework pool user priority number for  $u_{\text{low}}$  homework,  $1 \leq k \leq N$ . The  $M-N$  for homework pool user priority number for  $u_{\text{high}}$  homework,  $1 \leq l \leq M-N$ .

*Job Weight* formula of equation (6) into equation (7), after finishing the results as shown in equation (8).

$$\begin{cases} \frac{j_k}{\sum_{k=1}^N j_k + \sum_{l=1}^{M-N} j_l} \geq \frac{j_k}{\sum_{k=1}^N j_k + \frac{u_{\text{high}}}{u_{\text{low}}} \sum_{l=1}^{M-N} j_l} \\ \frac{j_l}{\sum_{k=1}^N j_k + \sum_{l=1}^{M-N} j_l} \leq \frac{j_l}{\frac{u_{\text{low}}}{u_{\text{high}}} \sum_{k=1}^N j_k + \sum_{l=1}^{M-N} j_l} \end{cases} \quad (8)$$

According to the equation (8) can be obtained and the relationship between such as the equation (9).

$$\begin{cases} \frac{u_{\text{high}}}{u_{\text{low}}} \geq \frac{\max(\sum_{l=1}^{M-N} j_l)}{\min(\sum_{l=1}^{M-N} j_l)} = \frac{(M-N) \times \max(j_l)}{(M-N) \times \min(j_l)} = \frac{4}{0.25} = 16 \\ \frac{u_{\text{low}}}{u_{\text{high}}} \leq \frac{\min(\sum_{k=1}^N j_k)}{\max(\sum_{k=1}^N j_k)} = \frac{N \times \min(j_k)}{N \times \max(j_k)} = \frac{0.25}{4} = \frac{1}{16} \end{cases} \quad (9)$$

The relationship of  $u_{\text{high}}$  and  $u_{\text{low}}$  meet  $u_{\text{high}} \geq 16 u_{\text{low}}$ , priority set with the user, the corresponding factors as shown in Table I.

Priority	Low	High
Priority Factor	1.0	16.0

In conclusion, based on user priorities and priority combination of scheduling weight calculation formula of the complete as shown in equation (10).

$$\text{jobWeight} = \begin{cases} u \times j \times \log_2(\text{tasksNum} + 1), \\ \text{considering jobsizes} \\ u \times j, \text{ otherwise} \end{cases} \quad (10)$$

where  $j$  and  $u$  values as shown in Table I.

#### 2) Based on time weight calculation threshold homework

Add a new user configuration file-weight-info. XML is used to set the user priority information and waiting time threshold BBBB. Equation (6) into equation (1), (2) and (3), after finishing the results are as follows:

When *Wait Time*  $\leq$  *Time Threshold*, weight calculation formula with the equation (11).

When *Wait Time*  $\geq$  *Time Threshold*, super users and ordinary users to submit homework, its weight calculation formula to satisfy equation (11).

$$\text{jobWeight} = \begin{cases} u_{\text{high}} \times j_{\text{max}} \times \log_2(\text{tasksNum} + 1), \\ \text{considering jobsizes} \\ u_{\text{high}} \times j_{\text{max}}, \text{ otherwise} \end{cases} \quad (11)$$

### III. THE ALGORITHM EXPERIMENT AND RESULT ANALYSIS

Set up the cluster Map capacity and Reduce capacity is 6. Assume various types call tracking data processing to take up the same cluster resources, set up three job pool has a weight of 1, set up the Map and Reduce the minimum of each homework pool Shared resources to 2. Research at the same time submit to the three assignments pool of different priorities, sent to the different priorities to submit homework pool assignments, the assignments pool resources usage. Specific experimental design as shown in Table II.

TABLE II: HOMEWORK POOL OF THE DATA TYPE BASED ON THE EXPERIMENTAL DESIGN

The serial number		Homework pool MR	Homework pool NMR	Homework pool default
1	Assignment priority	VERY_HIGH	NORMAL	VERY_LOW
2		NORMAL	VERY_LOW	VERY_HIGH
3		VERY_LOW	VERY_HIGH	NORMAL

According to the listed in Table II, the three groups of experiments, in each group of experiments, respectively to three homework pool each submit a job, and submit the three assignments of priority is different. Statistics of three groups of experiments three job running time, the Map task start and end time, and will be counted in the data mapping, the results such as shown in Fig. 1 and Fig. 2.

From Fig. 1 and Fig. 2 can draw the following conclusion: when the three assignments in the pool have assignments submitted at the same time, before meet the demand of this homework pool their resources, the job pool of resources is not Shared with other homework pool, and the allocation of resources between each homework pool also won't because the pool assignment is different priority. So the average share resources under the premise of three homework completion time is almost the same in the pool assignments.

Carried out in accordance with the listed in Table II, the

three groups of experiments, in each group of experiments, respectively to three homework pool each submit a job, and submit the three assignments of priority is different. Statistics of three groups of experiments three job running time, the Map task start and end time, the result data as shown in Table III.

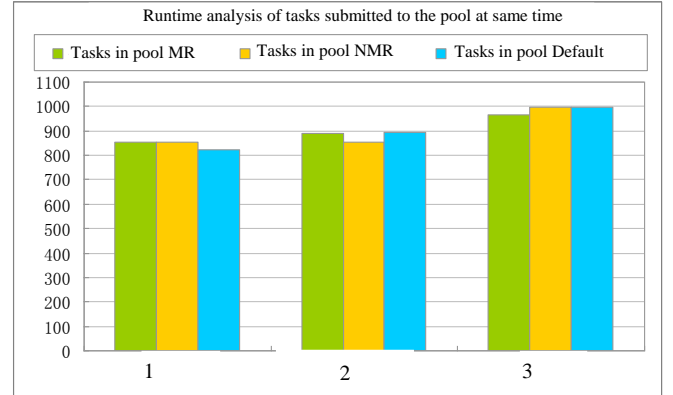


Fig. 1. Three homework assignments running time.

TABLE III: THE THREE ASSIGNMENTS OF THE AVERAGE RUN TIME OF OPERATION

The serial number		Homework pool MR	Homework pool NMR	Homework pool default
1	Average run time	588.6s	728s	699.6s
2		690.6s	808.6s	796.4s
3		866.8s	864.8s	867s

From Table III and Fig. 3 can draw the following conclusion: when the cluster is only one homework pool have homework, the homework pool will be reserved system resources, along with subsequent other pool assignments submitted, the first homework pool in his task completed release some resources, these resources will be shared with job submission and not reached the homework pool of resources sharing values. With the passage of time, all have homework assignments pool average tend to share the resources of the cluster.

### ACKNOWLEDGEMENTS

The research work was supported by National Natural Science Foundation of China under Grant No. 51479159.

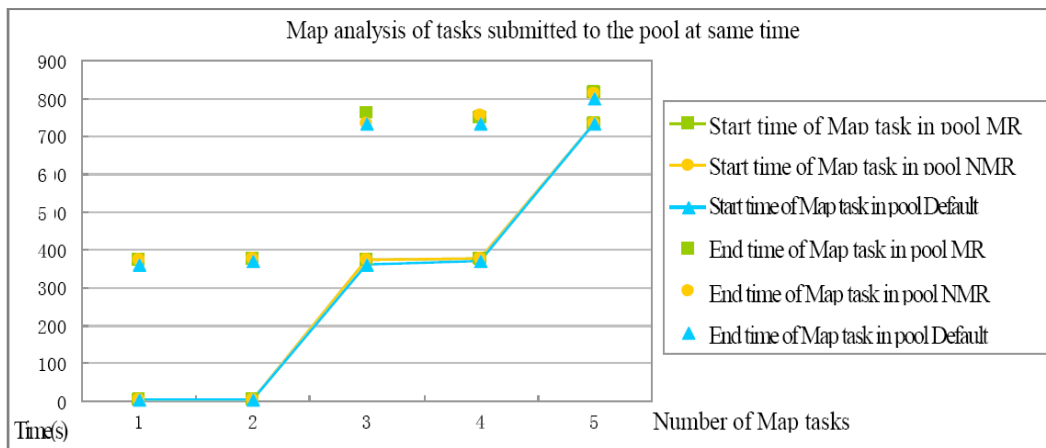


Fig. 2. Three homework assignments pool map task at the same time.

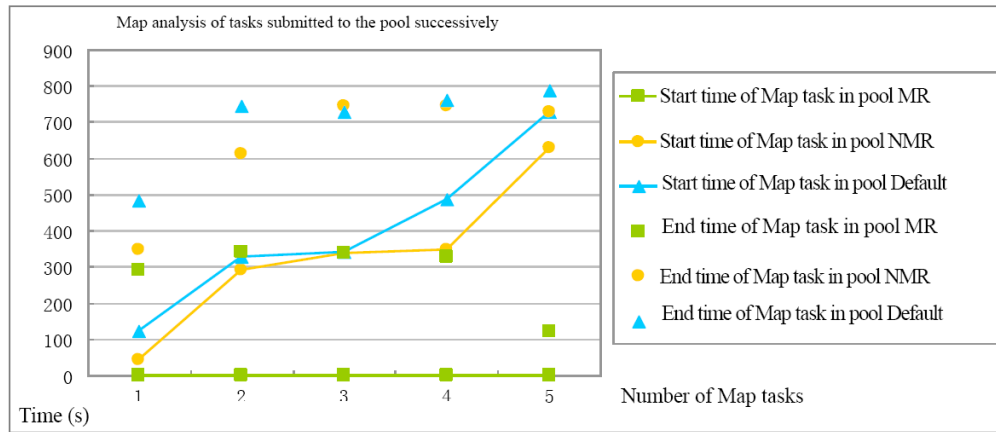


Fig. 3. Three homework assignments map task.

## REFERENCES

- [1] L. Liu, Y. Zhou, M. Liu, G. Xu, X. Chen, D. Fan, and Q. Wang, "Preemptive Hadoop jobs scheduling under a deadline," in *Proc. Eighth International Conference on Semantics, Knowledge and Grids*, 2012, pp. 72-79.
- [2] K. Kc and K. Anyanwu, "Scheduling Hadoop jobs to meet deadlines," in *Proc. IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, 2010, pp. 388-392.
- [3] M. Zaharia, D. Borthakur, J. S. Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling," in *Proc. 5th European conference on Computer systems*, 2010, pp. 265-278.
- [4] E. M. Mocanu, M. Florea, M. I. Andreica, and N. Tapus, "Cloud computing-task scheduling based on genetic algorithms," in *Proc. IEEE International Conference on Systems Conference (SysCon)*, 2012, pp. 1-6.
- [5] P. Lu, Y. C. Lee, C. Wang, B. B. Zhou, J. Chen, and A. Y. Zomaya, "Workload characteristic oriented scheduler for MapReduce," in *Proc. IEEE 18th International Conference on Parallel and Distributed Systems*, 2012, pp. 156-163.
- [6] P. Nguyen, T. Simon, M. Halem, D. Chapman, and Q. Le, "A Hybrid scheduling algorithm for data intensive workloads in a MapReduce environment," in *Proc. IEEE/ACM Fifth International Conference on Utility and Cloud Computing*, 2012, pp. 161-167.
- [7] S. J. Yang, Y. R. Chen, and Y. M. Hsieh, "Design dynamic data allocation scheduler to improve MapReduce performance in heterogeneous clouds," in *Proc. Ninth IEEE International Conference on e-Business Engineering*, 2012, pp. 265-270.
- [8] He, C., Y. Lu, and D. Swanson, "Matchmaking: A new MapReduce scheduling technique," in *Proc. Third IEEE International Conference on Cloud Computing Technology and Science*, 2011, pp. 40-47.
- [9] Y. Tao, Q. Zhang, L. Shi, and P. Chen, "Job scheduling optimization for multi-user MapReduce clusters," in *Proc. Fourth International Symposium on Parallel Architectures, Algorithms and Programming*, 2012, pp. 213-217.
- [10] M. Zaharia, D. Borthakur, J. S. Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Job scheduling for multi-user MapReduce clusters," Tech. Rep. UCB/EECS-2009-55, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 2009.
- [11] C. L. Abad, Y. Lu, and R. H. Campbell, "DARE: Adaptive data replication for efficient cluster scheduling," in *Proc. IEEE International Conference on Cluster Computing*, 2011, pp. 159-168.



**Qingnian Zhang** was born in Xi'an in 1957. He received his BS degree in design and manufacture of port machinery from Wuhan Institute of Water Transportation Engineering, China, in 1982 and a MS degree in transportation management engineering from Wuhan Transportation University, China, in 1988. He received his Ph.D. degree in machinery design and theories from the Wuhan University of Technology, China, in 2002. He has been a professor of the School of Transportation at Wuhan University of Technology since 2002. His current research interests include traffic and transportation planning, optimization and decision making of transportation system, transportation safety management etc.



management etc.

**Gao Ying** was born in Jilin in 1975. She received her BS degree in automobile application engineering from the Northeast Forestry University, China, in 1996 and received her MS degree in public management from Jilin University, China, in 2012. She is a Ph.D student in Wuhan University of Technology. Her current research interests include traffic and transportation planning, optimization and decision making of transportation system, transportation safety



Hadoop platform, optimization decision etc.

**Zhao Chen** was born in Shaanxi Province, China, in 1969. He received the B.S. degree in Chinese language and literature from China People's Police University, Beijing, China, in 1992, and MS degree in transportation management engineering from Wuhan University of Technology, China, in 2002. He is currently pursuing the Ph.D. degree in logistics management at Wuhan University of Technology. His current research includes parallel computing on



**Xiuxia Zhang** was born in Xuzhou in 1986. She received her BS degree in transportation engineering from Huaiyin Institute of Technology, China, in 2010 and MS degree in transportation planning and management from Wuhan University of Technology, China, in 2012. Her current research interests include traffic and transportation planning, optimization and decision making of transportation system.