

# A Parallel Task scheduling Algorithm Based on Fuzzy Clustering in Cloud Computing Environment

Qian Zhang, Hong Liang, and Yongshan Xing

**Abstract**—Parallel task scheduling is one of the core problems in the field of cloud computing research area, which mainly researches parallel scheduling problems in cloud computing environment by referring to the high performance computing required by massive oil seismic exploration data processing. Because of the natural reparability of Seismic data, it should maximize the use of computing resources to put the job file to the resource nodes, which can just meet the task computing requirements. This paper proposes scheduling optimization strategy of task and resource hybrid clustering based on fuzzy clustering, conducts the the clustering partition solution of concurrent job according to matching degree of task and resource nodes and narrows task scheduling scale and, narrows task scheduling scale and at the same time lays the foundation for dynamic acheduling tasks. After the division is completed, improved Bayesian classification algorithm is introduced to fast match tasks and computer according to real-time load and queue operations. In the end, verified by experiments, this scheme has higher efficiency.

**Index Terms**—Cloud computing, parallel scheduling, fuzzy clustering, task and resource hybrid clustering, Bayesian classification algorithm.

## I. INTRODUCTION

In the current mainstream distributed computing environment, the research on parallel scheduling [1] of jobs becomes a main direction to improve the working efficiency and one of the major research points of the field in the future. In cloud computing environment, including the homogeneous or heterogeneous multi-resource and multi-cluster environment, the computing capability of each resource node is generally unbalanced, job servers need take the execution sequence between nodes into consideration while processing parallel jobs and complete them within a certain time period as far as possible, which requires cloud computing environment has appropriate resource management model and excellent and reliable job scheduling strategy. In the paper, we research the parallel scheduling problems in cloud computing environment by the reference to the high performance computing required by massive oil seismic exploration data processing. The research aims at improving the efficiency of data processing, using cloud computing resources fully and efficiently, submitting tasks transparently, conveniently and efficiently, matching fast and executing them in optimal computing

resources efficiently.

Due to natural data separability [2] of seismic data, the link between data is limited, so relatively the efficiency of concurrent execution of a data file will be higher. So executing the job queue in the resource nodes which can meet the demand for computing capability can maximum use of existing computing resources. And due to the dynamic nature of the cloud environment, the attributes of resource nodes and the description of tasks' demand on resource requirements are also fuzzy. Therefore in the paper, we propose an algorithm of hybrid clustering of tasks and cloud computing resources to implement the task clustering. The features of the tasks to be scheduled will be extracted and analyzed in the algorithm, then the features of the cloud computing resource nodes will be extracted, and last the above two results will be fuzzy cluster analyzed. The nodes which meet the demand for task processing and have the best efficiency are divided into a group. We conduct clustering on different job queues and use the method to achieve the aim of scheduling optimization.

We use Bayesian classification algorithm to improve the task queue after the division. We match the resource nodes with the jobs in the queue based on real-time load fast. In the process of scheduling, the job scheduling is continuously improved by learning job attributes, the jobs on the server are conducted dynamically for adjusting parameter. A fast scheduling algorithm is implemented, suitable for the parallel execution of multiple job queues and able to complete the jobs correctly.

## II. OPTIMIZED PARTITION STRATEGY BASED ON FUZZY CLUSTERING

### A. Parallel Characteristics of Seismic Data

Seismic data has its own unique characteristics: (1)the data management of trace gathering is carried out according to the line, channel data, such as shot gather processing based on CDP, common receiving point(CRP), common shot point (CSP) and common midpoint (CMP); (2)when collected, original data is stored in the type of beam, shot, line, channel, etc. This type of data is suitable for concurrent execution; (3) the processing such as the seismic migration and velocity estimation need all sorts of transformation, Fourier, F-K and so on. If such kind of massive operations perform at the same time, parallel processing is the best choice, it can get very high processing efficiency. We can know from the above analysis of the characteristics, seismic data is more suitable for segmentation process; it can be executed parallel processing in shot gather and single shot.

Manuscript received March 5, 2014; revised May 25, 2014.

Zhang Qian is with College of Computer and Communication Engineering, University of Petroleum, China (e-mail: zhangqianupc@163.com).

The parallel mode the paper mainly researches parallel partitions the seismic data according to the number of shots in shot gather (which can be one or more shots). After dividing into multiple job queues, we can perform migration imaging processing of each shot data at maximum degree of concurrency simultaneously, and after dividing the jobs in this method we can find that each job queue does not need to communicate. Combined with parallel programming [3] Map Reduce, it can better adapt to the processing of cloud computing environment.

Based on the above analysis of the seismic data, and considering the specific cloud computing platform and parallel programming model used in the experimental environment, the following problems need to be solved when partitioning the seismic data:

1) Deconvolution, pre-stack (post-stack) depth (time) migration processing, imaging and interpretation and so on, are needed in the complete process of seismic data processing, and the amount of computational complexity in each step is large. If the master node directly submits the executed task, the communication between Map node and Reduce node or between each Map node would account for the major proportion of the system capacity and the major node would also be overloaded. Therefore, it is nontrivial to consider performing a proper partition for seismic data. In this paper, we adopt a processing method based on CSG parallel partition, which partitions each shot data as a subtask or several shots data as a subtask with a large amount of data. After the partition aforementioned, it would be loose coupling between each subtask. Hence, this way of task partitioning is similar to the loose coupling parallel scheduling of batch job.

2) In the specific of cloud computing environment, considering the principle of "birds of a feather flock together", we classify the nodes of almost equal performance, such as calculation and storage as a class, and then make every job queue actually face to an autonomous node region, which can also be considered as a child cloud environment of the overall cloud computing environment. This kind of processing mode can represent the advantage of the high-performance nodes processing complex tasks and low-performance nodes processing simple tasks in terms of the partition of jobs.

3) After partitioning the seismic data in specific cloud computing environment, the storage mode can base on sharing distributed storage HDFS. Through this method the data of all subtasks are centralized managed via Name Node. In the beginning of specific tasks, job servers will distribute the initial data according to the matching degree of the queue and resource nodes and assign the tasks to be processed to the node having data. Through partition and preparation, it will further reduce the huge cost of transmission and communication between Map and Reduce node.

### B. Algorithm Thought

Due to massive scale of seismic data, in order to optimize the task scheduling and avoid the waste of computing resources, this paper considers a task partitioning algorithm of hybrid clustering of tasks and cloud computing resources(CFCM). The features of the tasks to be scheduled

will be extracted and analyzed in the algorithm, then the features of the cloud computing resource nodes will be extracted, and last the above two results will be mixed according to some mathematical way and fuzzy cluster will be analyzed. The nodes which meet the demand for task processing and have the best efficiency are collected into a group. The nodes of low processing capacity are left to small jobs and the ones of high processing capacity are left to large jobs. We can use the method to achieve the aim of scheduling optimization.

The algorithm CFCM will express the jobs'(to be scheduled and processed) craving level of computing capability of resources as task vector and express the set of resource nodes' own attributes, such as its own computing capability as resource vector. The attributes of same location in vector have the same meaning, that is, they have the same dimension. Via standardized processing the above provided attribute vectors to standardized processing, the two attribute vectors are mixed together and perform clustering calculation, so the results of same hierarchical attribute obtained by clustering are divided into a group. Considering that all the factors influencing the partition of task would be too complex and any kind of partition can actually not deal with all the minor factors, the partitioning algorithm only extracts the main factor which strongly influences the partitioning result.

**Definition 1:** The related parameters of resource attribute in cloud computing environment such as the total computing capability, node number, node hard-disk space, node CPU number, dominant frequency, the size of the memory, network communication performance and the efficiency of the I/O are respectively expressed as AC, N, H, C, CF, M, T, IO, etc.; Task attributes such as the size of shot gather, the number of shot, the size of shot datum, the channel number of single shot data are respectively expressed as GS, GN, GD, TN.

**Definition 2:** The job vector is expressed as  $X = (x_1, x_2, \dots, x_s)$ ,  $s$  is the number of shot,  $x_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$  is the data vector representation of each shot in  $X$ ,  $x_{i1}, \dots, x_{im}$  is  $m$  index characteristics of each shot data  $x_i$ . In the vector,  $x_{im}$  represents the  $m$ th attribute index's initial data of the  $i$ th object to be partitioned in the job set. The definite parameter is shown in definition 1.

**Definition 3:** The resource vector is expressed as  $R = (r_1, r_2, \dots, r_n)$ ,  $n$  is the number of nodes,  $r_i = \{r_{i1}, r_{i2}, \dots, r_{im}\}$  is the data vector of each node,  $r_{i1}, \dots, r_{im}$  is  $m$  index characteristics of  $r_i$ . The definite parameter is shown in definition 1.

**Definition 4:** The weight vector is expressed as  $w = (w_1, w_2, \dots, w_m)$ ,

$$\sum_{i=0}^m w_i = 1 \quad (1)$$

$w_i$  is the parameter of weight vectors in different cloud computing environment, its value is different.

As shown in Fig. 1, hybrid clustering optimization algorithm of tasks and resources mainly includes the following four steps: (1) describe parameters preprocessing; (2) standardized processing; (3) mix data and vector of the task module data to establish fuzzy similar matrix; (4) execute clustering partition directly.

### C. Algorithm Steps

#### 1) Describe parameters preprocessing

The step mainly considers how to partition a new task. Under the circumstances of the known environment, we select the computing performance of nodes, computing performance tasks need, the size of space available and the size of task data and so on as the main influencing parameters of job partition stage. The theory is partitioned according to fuzzy matrix and each incidence degree converts to real-value interval [0, 1].

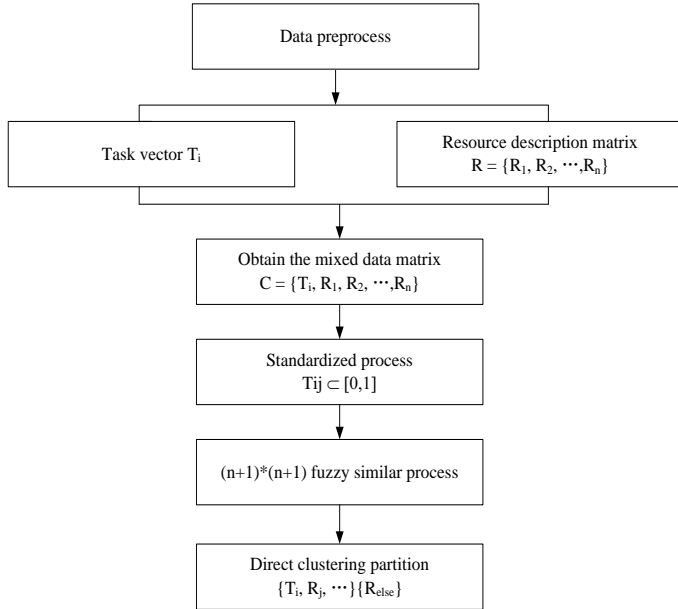


Fig. 1. Flowchart of the first task and resource hybrid clustering.

The computing performance of one node is expressed as

$$x_{1j} = \begin{cases} \alpha \times C \times CF + \beta \times M + o(x_{1j}), & j > 1 \\ w_1 \times TN, & j = 1 \end{cases} \quad (2)$$

$j > 1$  is the nodes oriented computing performance,  $j = 1$  is the nodes oriented computing performance  $\alpha, \beta$  is performance parameters,  $o(x_{1j})$  is other influencing part(can be neglected), only considering  $C, M, CF$ , respectively correspond to the number of CPU, the size of the memory and dominant frequency.  $TN$  is the channel number of shot data.  $w_1$  is the value of weight vector.

To avoid the situation of insufficient space occurring due to multi-replica node, the paper defines the size of the data as

$$x_{2j} = \begin{cases} H, & j > 1 \\ w_2 \times GD_k, & j = 1 \end{cases} \quad (3)$$

$H$  is the size of the  $j$ th node hard disk.  $GD_k$  is the size of shot data corresponding to shot task(the  $k$ th shot).  $w_2$  is the adjustable parameter of weight vector. If the data interaction is more, we can add the value of the parameter appropriately; otherwise, we can reduce the value.

The node number corresponding to shot number is expressed as

$$x_{3j} = \begin{cases} N, & j > 1 \\ w_3 \times GN, & j = 1 \end{cases} \quad (4)$$

$N$  is the node number in cloud computing environment,  $GN$  is the shot number in shot gather,  $w_3$  is the adjustable parameter of weight vector.

The total computing capability corresponding to the shot gather size is expressed as

$$x_{4j} = \begin{cases} \sum_{n=2}^{N+1} x_{1n}, & j > 1 \\ w_4 \times \sum_{k=1}^{GN} GD_k, & j = 1 \end{cases} \quad (5)$$

The computing data obtained above can only be taken as the computing performance representation in some single attribute. After further standardized processing, it can represent the specific meaning of its computing capability.

#### 2) Standardized processing

Convert  $x_{ij}$  to the value with weights of each parameter in the interval [0, 1] and make up the matrix  $X^{m \times (n+1)}$ . Usually there are several transformations, translation-standard deviation transformation, logarithmic transformation and translation-range transformation. Here we select the first standardized processing.

$$\bar{x}_{ij} = \frac{x_{ij} - x_j}{s_j}, (i = 1, 2, \dots, m; j = 1, 2, \dots, n+1) \quad (6)$$

$$\begin{cases} \bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij} \\ s_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2} \end{cases} \quad (7)$$

Through the above standardized processing, the standard deviation of the variable after transforming is 1, the mean is 0. And it reduces the effect of dimension to some extent. However, it cannot promise that the result  $\bar{x}_{ij}$  need go through the following processing in the interval [0, 1]:

$$x_{ij}^* = \frac{x_{ij}' - \min_{1 \leq i \leq n} \{x_{ij}'\}}{\max_{1 \leq i \leq n} \{x_{ij}'\} - \min_{1 \leq i \leq n} \{x_{ij}'\}}, (j = 1, 2, \dots, n+1) \quad (8)$$

We can obtain  $\forall i \in [1, m], \forall j \in [1, n+1], 0 \leq x_{ij}^* \leq 1$ , the effect of dimension is further reduced.

### 3) Establish fuzzy similar matrix using the mixed vector of the task data

1) Mix the requirement description vectors defined in definition 2 and 3 and get the mixed vector set  $\{T, R\} \rightarrow \{C_1, C_2, \dots, C_{n+1}\}$ . Each vector includes  $m$  influencing factors, that is  $\{T, R\} \rightarrow \{C_1, C_2, \dots, C_{n+1}\}$ . Then we can obtain the description matrix of tasks and resources.

In fuzzy clustering algorithm [4], the correlation coefficient between elements in fuzzy matrix usually includes angle cosine method, maximum and minimum method, arithmetic average minimum method and so on. The paper selects the arithmetic average minimum method, that is:

$$r_{ij} = \frac{2 \sum_{k=1}^m (x_{ik} \wedge x_{jk})}{\sum_{k=1}^m (x_{ik} + x_{jk})} \quad (9)$$

We can obtain  $r_{ij} \in [0, 1]$ .

2) We use the matrix to represent the fuzzy similar relation between each mixed vector.

$$R = [r_{ij}]_{(n+1) \times (n+1)} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1(n+1)} \\ r_{21} & r_{22} & \dots & r_{2(n+1)} \\ \vdots & \vdots & & \vdots \\ r_{(n+1)1} & r_{(n+1)2} & \dots & r_{(n+1)(n+1)} \end{bmatrix}_{(n+1) \times (n+1)} \quad (10)$$

The determination of the clustering number  $c$  is pointed by experts, but it lacks necessary credit when large-scale data is processed. We can adopt the method proposed by [5], which obtains the classification number  $c$  dynamically and adaptively. The best value interval of weighted index is [1.5, 2.5], and we usually select  $m=2$ .

### 4) Execute direct clustering classification

The paper adopts to the direct clustering classification algorithm to further improve the calculation speed. We no longer use massive calculation to seek the transitive closure  $t(R)$ , and abandon the complex seeking method [6] such as Boolean matrix method. After establishing the fuzzy similar matrix, we seek the classification of the task by setting the value of  $\lambda$  directly from its own, that is, complete a clustering.

To seek the clustering result of the job conveniently and avoid some exceptional cases such as the case that the task is classified into a class simply but the class has no resource node, the paper takes the attribute vector of each single shot job as the first column in the fuzzy matrix and fuzzy

clustering partitioning calculation method based on the equivalence relation in math. We can obtain the maximum in the first column using the calculation getting from a single ergodic, set it as the value of cutting level and perform clustering partition. Through adding the new job in the form of attribute vector repeatedly, we can classify the seismic data shot data and proper resource node into a class and obtain a new clustering result set  $R_s'$  of description job partition.

### D. Algorithm Flow

The job is set as  $JobID \in [1, s]$ , the node is set as  $RcID \in [1, n]$ , and the both are continuously numerical. As shown in Fig. 1, the steps of the hybrid clustering are as follows:

- 1) preprocess the data;
- 2) vectorizate the job  $\{job_1, job_2, \dots, job_s\}$  and vectorizate the resource  $\{re_1, re_2, \dots, re_n\}$
- 3) standardized process the initial matrix, the initial is  $k = 1$ ;
- 4) the initial is  $q = 0$ ;
- 5) compose the jobs to be scheduled and obtain the initial data matrix

$$U_k = \{job_k, re_1, re_2, \dots, re_n\}, k \in [1, s];$$

- 6) calculate the fuzzy similar matrix  $R_k = [r_{ij}]_{(n+1) \times (n+1)}$ ;
- 7) the clustering process: first determine the interception level, set  $\lambda = r_{21}$  and  $p[k][q] = 0$ ;

7.1) find the biggest  $\lambda$  vertically.

$$\text{for}(i = 2; i \leq n + 1; i++) \{$$

Extract the value more than  $\lambda$  in  $r_{i1}$  ergodically and record the position of  $i$   $p[k][q] = i$

$$q++;$$

7.2) find node which can run the job in the horizontal of the biggest  $\lambda$ .

$$\text{for}(i = 2; i \leq p[k][0]; i++) \{$$

If the horizontal exists a value more than  $\lambda$ , select it as the node number which classification can choose and record the position  $p[k][q] = i, q++$

$$k++;$$

8) If  $f(k \geq s)$ , all the partition of tasks and resources is completed, then turn to (9); otherwise, turn to the step (4) to continue executing next task partition;

9) Traverse  $p[k][q]$  and output the set of all the task queues matching the optimal node.

### III. THE SCHEDULING ALGORITHM BASED ON IMPROVED BAYESIAN CLASSIFICATION

#### A. Algorithm Thought

In the process of parallel processing computing tasks through MapReduce in the cloud environment, a job is divided into multiple tasks and parallel distributed to multiple nodes to perform the Map and Reduce operations at the same time. The role of the job scheduler is to arrange a task to run on a slave node reasonably. It should ensure the completion time as much as possible and avoid too much overload at the same time.

General Cloud scheduling algorithm preset the Map's number and Reduce's number of each slave node when the job is dispatched. Even on the premise of knowing the scale of tasks and the amount of computation, it is difficult to define excellent parameters without any experience and make the system complete the task efficiently. Setting the parameters too large or too small does nothing to help parallel programs obtain a shorter completion time. Especially for parallel tasks, FIFO parallelism is not high, and it obviously can't meet the basic processing need. Fair scheduling algorithms and computing capability need to set resource allocation and job queue partition. In the execution process the failure or task running time over the deadline can't guarantee the tasks completed as soon as possible [7]. Thus we can solve the problems by Bayesian scheduling algorithm. We can get the experience and achieve dynamic adjustment of the tasks on the task server by subsequent learning, instead of by presetting the parameters.

After Bayesian scheduling algorithm [8] divides jobs into good jobs and bad ones according to the load in a certain moment, jobs in good job queue will be forever scheduled and bad ones will be abandoned. This method does not consider the real-time load balancing of the system. It is possible that when the determined good job scheduling task is scheduled again after a moratorium, its results in system overload, that is, it does not consider the real-time load change of the Map and Reduce node. At the same time, the bad job may be a key to affect all the subsequent jobs. Terminating it may lead to the failure of the entire job.

The improved Bayesian scheduling algorithm needs to be combined with parallel job queue and the aforementioned hybrid clustering scheduling optimization algorithm, as shown in Fig. 2. Divide the jobs which have better concurrent granularity into multiple queue forms and allocate computing resources matching with the calculation amount of the job queue using the hybrid clustering optimization algorithm. When there is idle resource nodes, job servers will find its spare state according to the heartbeat information and choose one job from the longest job queue for classifying with the improved Bayesian scheduling algorithm. If the job is identified as a good job according to probability density, it starts scheduling execution; if it is a waiting job, it calculates the probability of the next job to be a good job or a waiting job until one appropriate job is found. During the specific execution we may encounter some queue which has no schedulable job. Then we need to transfer to the second longest queue to search schedulable

job. We can find the most suitable job running on it through the improved Bayesian job scheduling algorithm quickly.

In a particular embodiment, we need to combine hybrid clustering scheduling optimization algorithm to execute. Therefore first we consider whether the node of scheduling resources has slow tasks itself. If there are free nodes with other slow tasks, improved Bayesian scheduling algorithm does not need to perform, which can reduce remote data replication overhead. The idle node itself performs data backups and executes [9]. Completing the slow task assisted by a local resource node is faster than that by other nodes which need backup the data and then take the first parallel execution completed until the end of the mandate. If the above scheduling algorithm is applied to the data processing seismic data, in a way, it reduces the transmission of large amounts of data and communication overhead between different nodes in the backup and the ability to be able to Select the most computing tasks done quickly nodes corresponding to undertake the task of improving the efficiency of the algorithm.

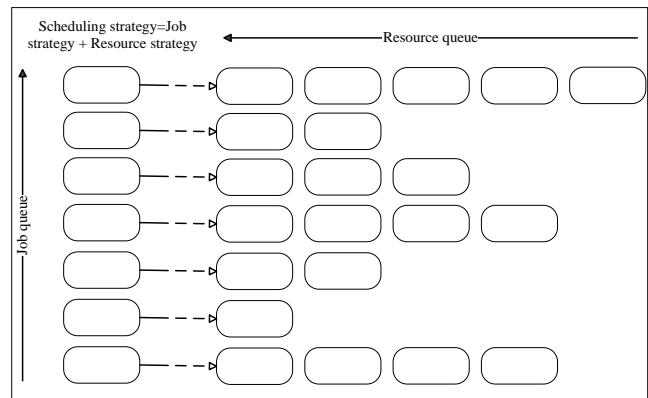


Fig. 2. The dynamic allocation map of jobs combining with resources.

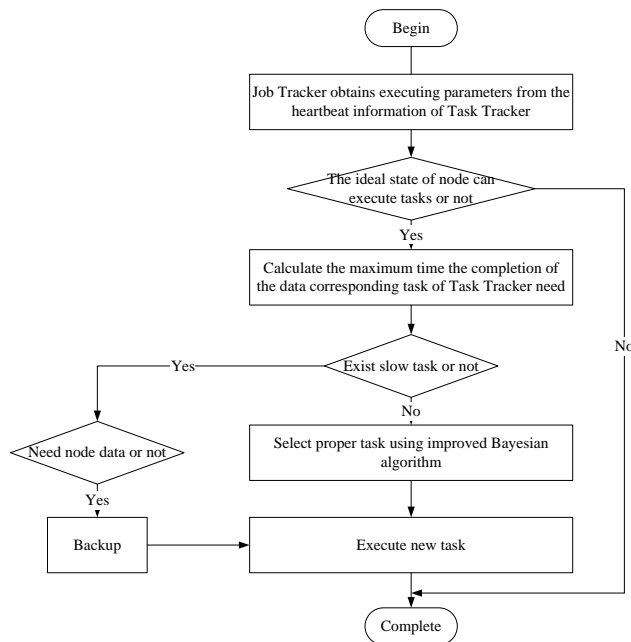


Fig. 3. parallel scheduling flow based on improved Bayesian algorithm

#### B. Algorithm Flow

From the thought description based on the Bayesian

improved scheduling algorithm in the above section<sup>[9]</sup>, this article improve the simple Bayesian classification algorithm and obtain the optimal job against specific nodes at some point. Through a simple process of learning or the set of default probability, setting the state monitoring value obtained in the overload and heartbeat message as the adjustment factor, the system can adaptively adjust the size of prior probability of different attribute values on the premise of different task classifications. Meanwhile, we can obtain the flow of the improved bayesian optimization algorithm by gathering optimized global scheduling strategy. It is shown in Fig. 3. We can see from Fig. 3, improved Bayesian scheduling optimized algorithm includes the following steps:

1) JobTracker obtains heartbeat information from the TaskTracker regularly. We can obtain the related parameters of tasks and nodes attribute from real-time heartbeat information, such as remaining CPU utilization of nodes, the remaining memory size;

2) Judge whether some node is spare or not by the parameters obtained from (1), if it isn't spare, do not need to execute the following operation; otherwise, call in the work, execute the step (3);

3) JobTracker periodically performs a calculation of all the currently executing tasks in the cluster according to the estimated time remaining, and then compares the task status of the node after calculation with the SlowTaskPoint of the slow task. If there is a slow task, the slow task is first executed; otherwise, execute step (4);

4) According to the scheduling optimization strategy, select the tasks relative to local node data preferentially from the two dimensional array of the queue to execute; If there are no data correlation jobs, after the partition, execute the step (5);

5) According to the improved Bayesian scheduling algorithm, find the job most suitable for execution using the maximum probability estimation, the steps is as follows:

5.1) The information which JobTracker accepts from the TaskTracker includes the attributes such as the remaining CPU utilization, the remaining memory size, the rate of I/O reading and writing. Combining with the size of jobs, the channel value, we can structure attribute variables corresponding to jobs' real-time state;

5.2) According to the set of overload conditions (the value of HighLoad), based on the probability of each attribute in the learning phase, we compare the completion of the tasks assigned to TaskTracker last time through the Bayesian classification scheduling strategy to regenerate the probability of each attribute in the latest state;

5.3) Choose a job from the longest waiting queue, calculate the maximum probability estimation, and determine the classification of jobs;

5.4) If the above calculated job is good, allocate the job to the resource nodes to be execute and complete scheduling in this stage, otherwise continue to execute (5.3) to choose job judgment;

6) After the completion of tasks in TaskTracker, the TaskTracker will sent the task status to JobTracker through

the heartbeat information. Repeat the step (1)-(6).

#### IV. TEST AND ANALYSIS OF THE EXPERIMENT

In this paper, the cloud computing environment used to test the results includes 8 nodes. Select a node as the major node, namely the directory server node (NameNode) of HDFS and the job server node of MapReduce, and the remaining nodes as working nodes (DataNode and TaskTracker). Each node is connected with gigabit switches. Four Dell PowerEdge R710 rack-mountable server are 4 core 2 CPU (Xeon E5520), the hard disk size is 500G, the cache is 4M, PC nodes are dual-core per CPU (Pentium E5), all nodes install RedHat Enterprise Linux 5.1, cloud computing system uses open-source Hadoop 0.20.0 to test the experiment and the operating system uses the RedHat Enterprise Linux 5.1.

##### A. The Experiment of Fuzzy Clustering Scheduling Optimization

In order to test the performance of static fuzzy clustering scheduling optimization, this paper set the computing capability and data size as the main reference factor. The dominant part of the task scheduling for computing ability, to define the weight vector  $w = (0.7, 0.7)$ , choose 200 mm data classified as a test case, and test job between concrete blocks and data processing of running time. Through observing the midway output of fuzzy clustering partition, we can know the job is divided into six classes (i.e., 6 queues). In order to compare with it, we artificially set data to multiple classifications.

After partitioning the job, we use the default FIFO scheduling method of MapReduce to process. The task execution time under the various partitions is as shown in Fig. 4.

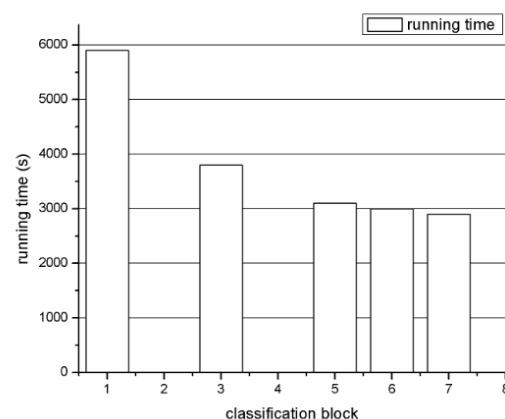


Fig. 4. The running time figure of fuzzy clustering job partition.

As we can see from the figure, the fuzzy clustering partition algorithm can find out more suitable coarse-grained classification method based on tasks and environmental levels corresponding characteristics and make preparation for the dynamic scheduling tasks of MapReduce according to the load. It is the basic of improving the overall execution efficiency of processing large-scale data in the cloud computing environment.



### B. Improved Bayesian Scheduling Algorithm Experiment

The improved Bayesian scheduling algorithm proposed in this paper and the other three kinds of scheduling algorithms commonly used in MapReduce are taken as an experiment for performance comparison. Among them, the FIFO scheduling algorithm is relatively easy, while the computing capacity scheduling algorithm and fair share scheduling algorithm are relatively difficult. Because they have to set the maximum of tasks that can be run in the server at the same time, what's more, the description of resources also need to increase one by one, in this way not only the workload and difficulty of the submitter will be increased, the overall operation performance may also be affected if an improper parameter is set. So in the specific experiment we need to set the parameter method more suitable for occupying CPU, reducing the effect of the performance of computing capacity scheduling algorithm and the fair scheduling algorithm.

In the specific experiment, we adopt global array to store resources and allocate nodes statically. After simply learning, we can set attributes (i.e. the possibility in good operation and waiting operation.), calculate 600,400,300 and 200 shots of seismic data and compare the results respectively. Fig. 5 clearly shows that it is approximately to be a straight line for FIFO scheduling algorithm[10], and its time is relatively stable comparing with similar task. However, the scheduling algorithm is not efficient. As to computing capacity scheduling algorithm and fairness scheduling algorithm, their operating time differs in light of the different numbers of tasks the server undertakes simultaneously and different settings of job queues. They display their own advantages when operating with different data because of different workload. There is a proactive process of learning in the improved algorithm we proposed, which makes it unreasonable overall and learning account for a larger amount of time than operating in the situation where the amount of gather data is little and no slow task in local nodes, however the operating time shows its advantage comparing to the other three algorithms as the data increases.

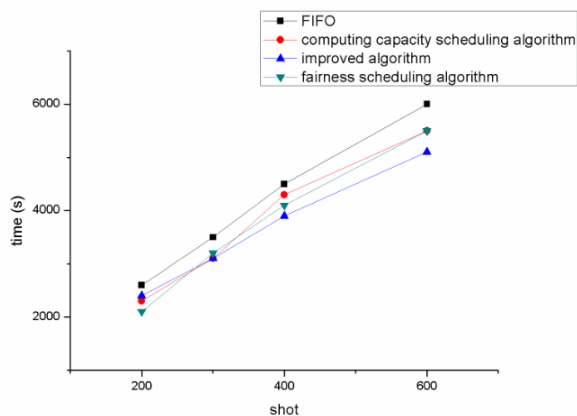


Fig. 5. Comparison of different scheduling algorithm handling shot data.

### V. CONCLUSIONS

This paper is carried out based on the processing of seismic data in the cloud computing environment. In the process of specific scheduling decision, we propose a

scheduling optimization strategy based on fuzzy clustering, in which, we dynamically combine the resource node information with job attributes, allocate resources meeting the demand of task requirement to the corresponding task and avoid scheduling tasks to the resource nodes which differ greatly. All resource nodes have the opportunity to be scheduled so as to achieve the good effect of load balancing. Based on the probability estimation of the Bayesian classification strategy, we introduce improved scheduling strategy, establish the collaborative relationship between the jobs and real-time node load, adaptively adjust attribute probability and schedule priority tasks most suitable for execution. Finally through the experiment we test the improved scheduling algorithm and compare it with the classical scheduling algorithm. The experimental result shows that after statically partitioning the seismic data, combining the partitioned jobs with fuzzy clustering optimization strategy and implementing dynamic scheduling through improved Bayesian classification scheduling algorithm can achieve the aim of faster computation efficiency.

### REFERENCES

- [1] Y.-H. Luo, J.-P. Liu, and G.-Z. Yu, "A review on the pertrack depth migration," *Geophysical and Geochemical Exploration*, vol. 28, pp. 540-546, Dec. 2004.
- [2] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Communications of the ACM*, pp. 107-109, vol.1, 2008.
- [3] W. Li and Q. Zhang, "Cloud scheduling algorithm based on fuzzy clustering," *Journal on Communication*, vol. 33, pp. 146-153, March 2012.
- [4] B. Li, "An improved fuzzy C-Means algorithm," *Computer Application and Software*, vol. 25, pp. 21-25, May 2009.
- [5] Y. Xia, "Research and improvement of Job scheduling algorithms in Hadoop platform," Ph.D. dissertation, South China University of Technology, Guangzhou, 2010
- [6] Z. Yu, "Research on improving Hadoop job scheduling based on learning approach," *Computer Science*, vol. 39, pp. 220-224, May 2012.
- [7] D. J. Hand, *Ten algorithms of machine learning: Naïve Bayesian*, Taylor & Francis Group, 2009, ch. 2, pp. 163-178.
- [8] S. Zhang, S. Zhang, X. Chen, and S. Wu, "Analysis and research of cloud computing system instance," in *Proc. 2010 Second International Conference on Future Networks*, pp. 88-92, vol. 26, 2010.
- [9] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A comparative study into distributed load balancing algorithms for cloud computing," in *Proc. the 24th IEEE International Conference on Advanced Information Networking and Applications Workshops*, pp. 551-556, Fukuoka, Japan, 2011.
- [10] J. Hu, J. Gu, G. Sun, and T. Zhao, "Scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in *Proc. the 3rd International Symposium on Parallel Architectures, Algorithms and Programming*, pp. 89-96, Liaoning, China, 2010.



**Zhang Qian** was born in July 1982, and in Dongying, Shan Dong province of China, serves as a lecture in the China University of Petroleum (Eatern China). In July, 2003, Zhang Qian graduated from College of Computer and Communication Engineering in China University of Petroleum and earned her Engineering bachelor degree. In July, 2007, she graduated from College of Computer and Communication Engineering in China University of Petroleum and earned her Engineering mater degree.

From July, 2006, she serves as a lecturer in the College of Computer and Communication Engineering in China University of Petroleum (Eatern

China). Her major researching field is in the full-fledged computing and cloud computing. She has participated in four provincial science projects, presided over one project funded by fundamental research funds for China's central top university, won two University Teaching awards and published fifteen papers as the first author, four of which are included in EI/ISTP.

Zhang Qian has participated in the Seismic Grid Technology Integration Software Research, which is a major project in Sinopec, and she ranks in the fifth position in fifteen members of this research. She participated in Applied Grid Technology Based on the Seismic Data Processing, which is funded by Petro China Youth Innovation funds and she ranks in the fourth position in ten members of this research. Her participated Grid

Middleware Key Technology Research and Development of GIS is finished and she ranks in the fourth position in nine members in this research and her participated Oil Disciplines Shared Grid Platform is under research and she ranks in the fourth position in eight members in this research. In addition, Lecture Zhang, has presided over lateral research project Current River Oil Production Plant Management System, College Youth Fund project GT-based dynamic grid resource scheduler and her presided University innovation fund project Cloud computing scheduling strategy based on resource aggregation is under research.