A cloud task scheduling algorithm based on QK-mean clustering

Guozhu Zhao, Liang Ma *, Ruibin Zhao

School of Computing & Information, Chuzhou University, Chuzhou Anhui 239012, China

Received 1 June 2014, www.cmnt.lv

Abstract

When classifying resources in cloud computing environment using the idea of clustering, the information entropy of resources' attribute can reflect the degree of significance in clustering process. Using information entropy, a task scheduling algorithm based on QK-mean clustering is proposed. We calculate the degree of significance of cloud resources' attributes, then apply K-mean clustering algorithm to classify the cloud resources according to the degree of significance of attributes, and we create Resources K-tree to store the process and result of the clustering. In this way, we transform the task scheduling process into the process of searching a suitable leaf node in Resources K-tree. The experimental results show that the QK-mean scheduling algorithm can effectively improve the efficiency of cloud task scheduling.

Keywords: task scheduling, information entropy, K-mean, cloud computing

1 Introduction

Cloud computing are rapidly gaining in popularity. Virtualization and grid computing technologies is applied to transform the physical resources into virtual resources, the cloud operator manager the resource by allowing the user to rent, only at the time when needed, only a desired amount of computing resources out of a huge mass of distributed computing resources without worrying about the locations or internal structures of these resources. The popularity of cloud computing owes to the increase in the network speed, and to the fact that virtualization and grid computing technologies have become commercially available. It is anticipated that enterprises will accelerate their migration from building and owning their own systems to renting cloud computing services because cloud computing services are easy to use, and can reduce both business costs and environmental loads. With the popularity of cloud computing, the task scheduling becomes more and more important. How should we schedule a task to cloud resources becomes a problem we have to focus on.

Lots of work has been done applying K-mean clustering or information entropy. Taking into account the background knowledge [1] of clustering objects, constraints that limit some object data must be divided in the same class or must not be divided in the same class are increased when K-mean cluster algorithm is applied. In the process of K-mean cluster algorithm, these constrains are checked and the result of the cluster must not break the rules constrains provide. K-mean algorithm takes high computational cost and the clustering results mostly dependent on the centers of initial clustering [2], researchers have proposed a number of ways to improve the K-mean algorithm. In order to improve the efficiency of clustering, the idea of MapReduce is thought to be introduced in the K-mean clustering in the reference [3], parallel K-mean clustering algorithm is proposed based on MapReduce. Information entropy [4] can solve the problem of quantitative information, Elimination of uncertainties of the thing, references [5-10] apply information entropy idea to cluster analysis, and achieves good effect. On the basis of these studies, we proposes QK-mean task scheduling algorithm. The algorithm uses the tools of information entropy to obtain the significance of cloud resource attributes, according to the significance of cloud resource attributes cluster the cloud resources using K-mean clustering algorithm repeatedly and create Resource K-tree, in the Resource K-tree formed in the process of QK-mean, selects suitable resources assemble for the task in the cloud environment. In this assemble select resources server for the task using Min-min algorithm.

Our contributions can be summarized as follows.

Firstly, we designed an algorithm to support our analysis. Based on the thoughts of K-mean clustering and information entropy, we take into account the significance (this process is calculated according to the information entropy of each attribute) of attributes of cloud resources, propose QK-mean algorithm, and apply the algorithm to the clustering of cloud resources. In fact, the process of QK-mean clustering algorithm creates a Resource K-tree by multi K-mean clustering according to the significance of attributes of resources, by this way the cloud resources can be classified into a series of classes shown as the leaf nodes (classes of resources) in Resource K-tree. Then we put forward scheduling algorithm based on the thoughts of QK-mean clustering further more: when a task comes, the Resource K-tree will be travelled to find a suitable leaf node in Resource K-tree, the task will be scheduled to the class and Min-min algorithm will be used in the class to select resources to server the task.

* ***Corresponding author's*** e-mail: ml@chzu.edu.cn

Secondly, we create a cloud environment based on Openstack to verify the performance of the scheduling algorithm based on QK-mean clustering algorithm. Sixteen physical machines was virtualized applying KVM virtualization technology in Openstack cloud environment. In the cloud, we maintain a proxy server to create and store the Resource K-tree, at the same time the proxy server decides which VMs server the certain task according to the result produced by the QK-mean scheduling algorithm, and schedules the task to the VMs. To evaluate the performance of QK-mean scheduling algorithm, we compare QK-mean scheduling algorithm to K-mean algorithm and Min-min algorithm in Openstack cloud environment.

Thirdly, we raised some other factors we should take into consider during our research. In view of the complexity and the feasibility of experiment, in this paper we mainly focus on the resources' three attributes: computing power, bandwidth and storage capacity. The attributes of cloud resources is various, for example the reliability, the cost, and so on. These factors will be considered in our following work.

2 The cloud task model and cloud resources model

Cloud tasks can be expressed as a graph (DAG): $G=(T,E)$. Among them, $T=\{t_0,t_1,t_2,...,t_n\}$ represents the task set, $n=|T|$ represents the number of tasks; E represents the edge set task execution preorder relations. the task t_i contains a lot of attributes, such as the uniquely identifies ID of task, the length of task, the state of task ,the computing power, bandwidth and needs of storage capacity, the cloud resources the task has access to, the latest completion time of task, the required data of task input and output.

The resource collection of cloud system can be expressed as $R=\{r_0,r_1,r_2,...,r_m\}$, $m=|R|$ is the number of resources. Any resource r_i $(i\in[0,m\text{-}1])$ of the resource set contained a lot of attributes, such as the unique identifier for the resource, the resource provider, the computing power resources, bandwidth, storage capacity and reliability of resources.

As can be seen, there is a demand of the cloud task for computing capacity of resource, bandwidth and storage capacity , and cloud resources itself has the attributes of computing power, bandwidth and storage capacity and etc. This article assumes the task of needs and resources of cloud resources' attributes itself has the attributes is one to one, clouds task requires the attribute IS, resources have the attribute $\mathbb{IS}$, $S\in\{1,2,3,...,n\}$, n is the number of attributes.

3 Resource clustering

3.1 THE SIGNIFICANCE OF ATTRIBUTE

Definition 1. R_P is defined as a resource attribute matrix, R_P can be expressed as:

$$R_p=\begin{bmatrix} r_{11},r_{12},r_{13},...,r_{1n} \\ r_{21},r_{22},r_{23},...,r_{2n} \\ \cdots \\ r_{m1},r_{m2},r_{m3},...,r_{mn} \end{bmatrix}, \tag{1}$$

wherein, $i\in[1,m]$, m is the number of resources; $j\in[1,n]$), n is the number of attributes of the resource, the matrix elements r_{ij} represents the attribute j of the i-th resource. As can be seen, the resources matrix is a matrix of m rows and n columns.

In the cloud, the quantity of information of resources' attributes include is different. In information theory, information entropy $H(x)$ is used to measure the disordered degree of systems. The expression of Entropy $H(x)$ [11-12] is shown in Equation (2).

$$H(x)=-\sum_{i=1}^{n}p(x_i)\ln p(x_i)\,. \tag{2}$$

Information is a measure of the ordering degree of system, the greater the difference between the value of attributes of cloud resources, the smaller information entropy. The greater the amount of information provided by the attribute, the greater the weight of the attribute; conversely, the smaller the difference between the value of attributes of cloud resources, the greater information entropy, the smaller the amount of information provided by the attribute, the smaller the weight of the attribute.

Definition 2. Define entropy of attribute j of cloud resources for e_j

$$e_j=-k\sum_{i=1}^{m}p_{ij}\ln p_{ij}6666666\,, \tag{3}$$

where p_{ij} represents the proportion of the value of attribute of j item accounts for the sum of attribute value of the i-th cloud resource, p_{ij} can be expressed as:

$$p_{ij}=\frac{r_{ij}}{\sum_{i=1}^{m}r_{ij}}\,. \tag{4}$$

Definition 3. In order to quantify the significance of the attributes of cloud resources, we define the significance of attributes of cloud resources as I_j :

$$I_j = \frac{g_j}{\sum_{j=1}^{n} g_j}, \tag{5}$$

where $g_j = 1 - e_j$, e_j is the information entropy of attribute j of cloud resources. Obviously, for attribute j, the smaller of the difference among r_{ij}, the greater the e_j, when all r_{ij} are equal, $e_j = e_{max} = 1$. In the case, attributes x_j effect little on the cloud resources clustering; the greater the difference among the attribute value, the smaller the e_j, and the greater the role played by attributes x_j of cloud resource clustering.

3.2 RESOURCE K-TREE

Assuming the attributes in the matrix of attributes are $I1, I2, I3, ..., IS$ and the attributes are ordered by attributes' significance according to the descending order, in which $s \in [1, n]$. When performing K-mean clustering of resources, consider only one attribute each time, namely, the first K-mean clustering only consider the attribute $I1$ with the greatest significance, after the clustering, cloud resources are divided into k classes according to the attribute $I1$. For each of these collections of class k, divide resources into k classes using K-mean clustering according to the attribute $I2$ of the second significance. And so on, through the n-th K-mean clustering, the resources are divided into k^n classes, and all the process of resources classification constitutes a complete k-tree, this tree is named Resource K-tree in the paper. For example, take $k = 2$ (the number of clusters for each category is 2), the attributes of cloud resource considered are computing power, bandwidth, storage capacity of resources (the number of attributes of resources equal to 3). Therefore, the i-th resource in the set of resources R can be expressed as $r_i = \{r_{iComp}, r_{iBW}, r_{iStor}\}$, where r_{iComp}, r_{iBW}, r_{iStor} denote computing capacity, communication capability and storage capacity of the i-th resource, Let's assume the 3 attributers' significance in the matrix of task attribute are I_{Comp}, I_{BW} and I_{Stor} according to the descending. For the first clustering, consider only the significance of the greatest I_{Comp} and divide cloud resources into 2 collections in accordance with the attribute I_{Comp}. After clustering, resources are divided into 2 collections according to attribute I_{Comp}, for each of the 2 collections, divide resources into 2 classes according to the attribute I_{BW} of the second significance. So on, through 3 K-mean clustering(3 is the number of attributes of resources), the resources are divided into k^3 classes, the complete k-tree constituted by l the process of clustering is showed in Figure 1, in which, R_{pq} (q-th nodes on layer pc in k-tree of resource) represents the sub-resource collection, C_{pq} represents the centre of clustering of R_{pq}.

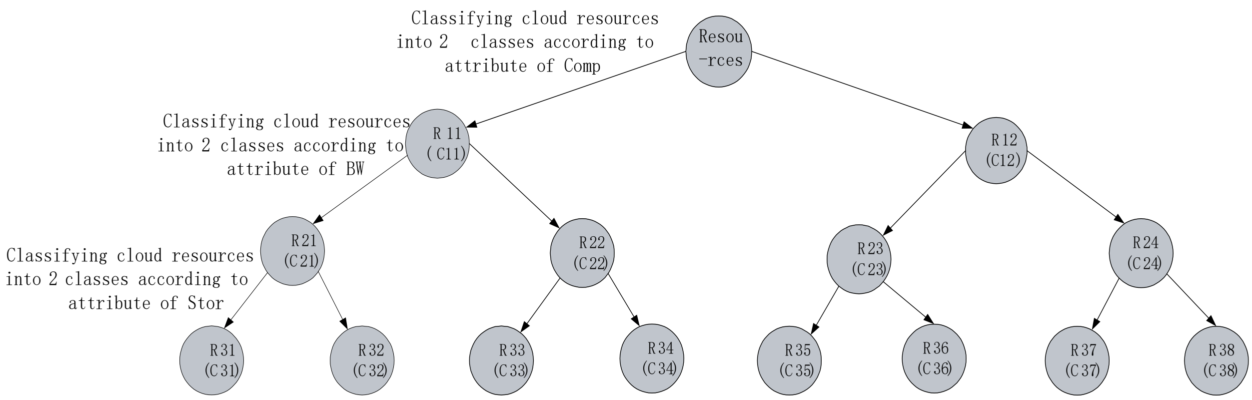

FIGURE 1 Binary tree of resources

4 QK-mean task scheduling algorithm

QK-mean task scheduling algorithm creates K-tree of resources based on the attributes of cloud resources which cloud providers and users are concerned about. When cloud task t comes, compare the distance between attributes $I1$ ($I1$ is the greatest important degree of cloud resources) and k centres of sub-class clustering. Schedule the task into the sub-category in which the distance between attribute value $I1$ and k cluster centres is the smallest, and so on, until the task is scheduled to leaf nodes, use Min-min algorithm in the subclass of the leaf node to select resources serving the task.

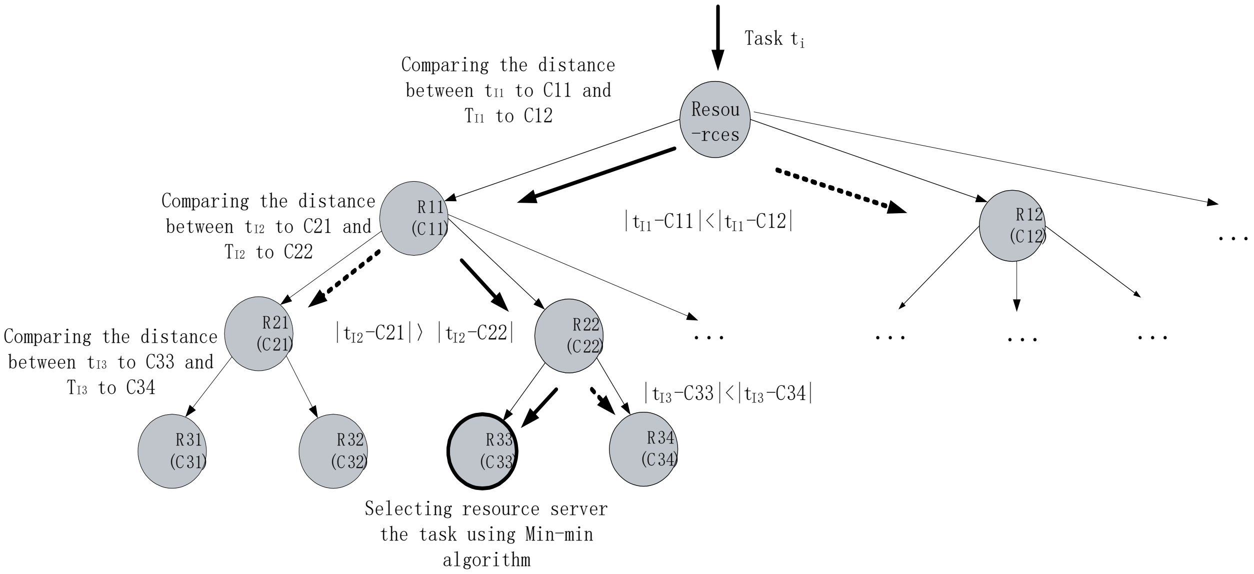

FIGURE 2 Task scheduling process of QK-mean task scheduling algorithm

As can be seen, the core idea of QK-mean task scheduling algorithm is to transform the task scheduling process into the process of searching a leaf node in Resource K-tree. As shown in Figure 2, firstly, the algorithm determines the scheduling of t_i is to find resources from R11 or from R12 based on attribute value t_{iComp} (Let's assume that the attribute value is the demand for computing power of the resource t_{iComp}). If $|t_{iComp}-C12| >= |t_{iComp}-C11|$ (C11 , C12 is the cluster centre of subclass R11 and R12), indicates the distance from computing power of resource demand for task to cluster centre of subclass R11 is shorter. The scheduling of t_i is to search resources from the sub tree where R11 is in, otherwise, the scheduling needs to search resources from the sub tree where R12 is in; so on, until the class needed to search for task t_i is a leaf node of the Resource K-tree. The resources collections of the leaf node is the most suitable for providing services for task t_i , so then select resources in the collections to provide services for the task using the Min-min algorithm.

4.1 PSEUDO-CODE OF THE ALGORITHM

According to the idea of QK-mean algorithm, QK-mean algorithm can be descripted available with the follow pseudo-code.

4.2 ENVIRONMENT OF ALGORITHM IMPLEMENTATION

OpenStack is a global collaboration of developers and cloud computing technologists producing the ubiquitous open source cloud computing platform for public and private cloud [13,14]. The project aims to deliver solutions for all types of clouds by being simple to implement, massively scalable, and feature rich. The technology consists of a series of interrelated projects delivering various components for a cloud infrastructure solution [15-16].

TABLE 1 Physical machine of Openstack

Node	Hardware	Number	Function
Cloud controller	Disk space: 300 GB (SATA) Volume storage: two disks with 2 TB (SATA) for volumes attached to the compute nodes Network: one 1 GB Network Interface Card (NIC)	1	runs network, volume, API, scheduler and image services of Openstack
Compute nodes	Processor: 64-bit x86 Memory: 32 GB RAM Disk space: 300 GB (SATA) Network: two 1 GB NICs	15	runs VMs

In this paper, the physical machine as shown in Table 1 is provided to simulate cloud computing environment. Based on Openstack, the architecture of the experiment is shown as Figure 3. These physical machines are visualized to serials of VMs or instances which represent cloud resources applying the technology called KVM used by Openstack in cloud computing environment. The cloud resources are classified and the cloud tasks are scheduled using the algorithms in Proxy server:

```
///////////Creating a Resource K-tree   named T///////////////
Creating attribute matrix of resources Rm*n
Compute significance of  attribute and sort them by the significance : I1, I2, ... In
Initial the K-ary tree T
//The initial state
T->data=V,T->Child1=NULL,T->Child2=NULL,…,T->Childk=NULL.
CreateRT（T）// T is the root of the tree
for each I{I1,I2,…,In}
   {
      for each vϵV// V is the set of nodes on i-layer of the tree T)
         { for resource set v, devide resources into k classes using K-mean
           algorithm according to attribute I1
         If(Ti is not a leaf node)
         {
         CreateRT（T1）
         CreateRT（T2）
         …
        CreateRT（Tu）// u is the number of non-leaf nodes
         }
      }
}
////////////// QK-mean scheduling algorithm /////////////////////
Input：T={t1,t2,…,tn}, Resource K-tree named  TR
QK-mean(Tree TR)
{
If(TR->Rchild!=NULL) // If the resource node is not a leaf node
    {
For each tiT do
   For each IjI do
      traverse K-tree TR breadth
    Cj=Min{tij-C}
   QK-mean (TRcj)
    }
    Else
    {
     Return C //Return a collection of leaf nodes
     Ti->C //schedule task ti to resource collection C
     Min-min(ti,C) //select a virtual machine for task ti in resource  using
Min-min algorithm
     ti->VM  //schedule ti to the specified virtual machine
    }
}
```

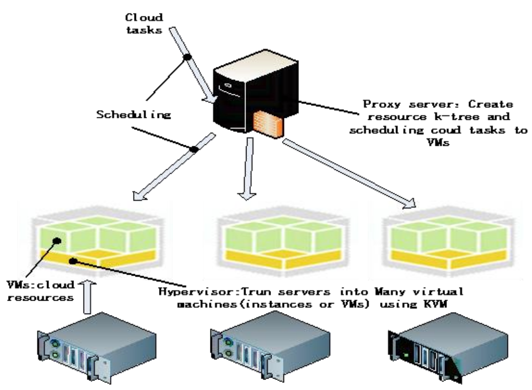

FIGURE 3 Cloud environment based on Openstack

So before task scheduling, cloud resources are classified in advance in accordance with the proposed method. When task t_i comes, proxy server get all of attributes of the task and traverses Resource K-tree, deciding which VMs (cloud resources) t_i is scheduled to.

5 Experimental design and experimental results

5.1 EXPERIMENTAL DESIGN

This paper considers the resources' attributes of computing power, bandwidth and storage capacity, Take k equal to 3, K-Mean algorithm, Min-min algorithm [17-19] and QK-mean algorithm are used to schedule the same tasks respectively. 2000 cloud tasks are scheduled to the 20 resource nodes (virtual machines), the cloud tasks requirements for computing power, bandwidth and storage capacity of the resource are generated randomly. To quantify the computing power, bandwidth and storage capacity of cloud resources, take the calculation of computing power for example. In this paper, the resource with the worst computing power in the Openstack cloud is set to be scalar quantity represented as c_{min}, Computing power of the i-th resource R_{ic} can be represent as $R_{ic} = c_i / c_{min}$, c_i represents the computing power of the i-th resources, So the methods calculate the capacity of bandwidth and storage is identical. The parameter of computing power, bandwidth and storage capacity of the 20 cloud resource is shown in Table 2.

TABLE 2 Resource performance parameters

Resource ID	Computing power	Bandwidth	Storage capacity
1	29	71	91
2	27	42	33
3	85	45	63
4	88	47	51
5	87	61	40
6	35	18	51
7	57	17	17
8	16	79	52
9	59	56	44
10	31	21	30
11	66	10	84
12	86	16	80
13	57	41	70
14	99	42	36
15	79	15	18
16	15	94	82
17	73	5	19
18	19	38	45
19	64	31	90
20	57	66	31

In order to ensure the results of comparison of algorithm scientific, task scheduling algorithm based on K-mean clustering is designed to compare with QK-mean algorithm. Performance of resources is weighted by the resource attribute values K-mean clustering algorithm, Weights value of resource attribute is equal to the attribute' significance calculated in 3.1 of this paper. Performance of cloud resources is represented as r_{GP_i} witch is calculated as follows:

$$r_{GP_i} = \sqrt{\frac{\sum_{j=1}^{k} I_j r_{ij}^2}{\sum_{j=1}^{k} I_j}}, \tag{6}$$

r_{ij} represents the i-th resource j-th attribute, I_j represents the significance of j-th attribute. The distance between r_p and r_q can be represented as rd_{pq} and can be calculated as follows:

$$rd_{pq} = \left| r_{GP_p} - r_{GP_q} \right|. \tag{7}$$

The cloud resources can be Classified into k classes whose clustering centers can be represented as $C_1, C_2, ..., C_k$ based on K-mean clustering algorithm, The i-th task demand for resources can be represented as $rq_i = \{rq_0, rq_1, rq_2, \ldots, rq_m\}$. The performance task demands on resources can be represented as r_{Gqi}:

$$r_{GQ_i} = \sqrt{\frac{\sum_{j=1}^{k} rq_{ij}^2}{k}}. \tag{8}$$

i-th task is scheduled to the class C_{select} that can be represented as $C_{select} = \mathrm{Min}\,\{|r_{GQ_i} - C_j|\}$, Min-min algorithm is used to select suitable resources in class C_{select} for the task.

5.2 ANALYSIS OF EXPERIMENTAL RESULTS

The 2000 cloud tasks are scheduled to 20 nodes use K-mean algorithm, Min-min algorithm and QK-mean algorithm respectively, the task completion time are shown in Figure 4, customer satisfaction are shown in Figure 5.

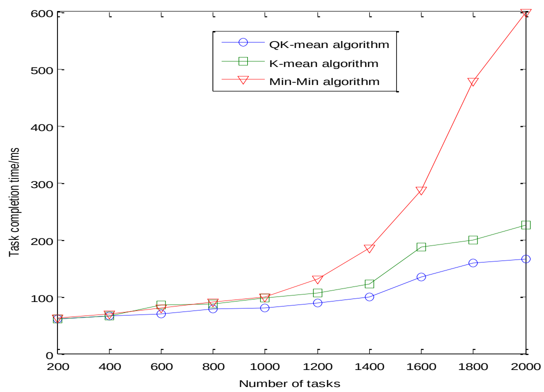

FIGURE 4 Clouds task completion time comparison

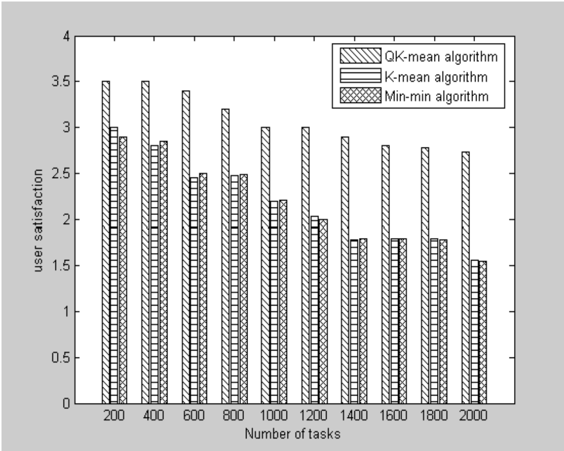

FIGURE 5 Comparison of user satisfaction

As can be seen, the significance of resource attributes are sorted firstly in accordance with the significance of the attributes of cloud resources in QK-mean algorithm. By search resources from Resource K-tree, reducing the size of the resources' selection, QK-mean task scheduling algorithm transforms the process of resource scheduling into the process of searching suitable leaf nodes in the Resource K-tree. When the task is scheduled to a leaf node(a subclasses in the leaf node of Resource K-tree) of Resource K-tree, Min-min algorithm is applied to select resources server for the task. So comparing to K-mean algorithm and Min-min algorithm, QK-mean algorithm can get smaller task completion time and better customer satisfaction.

6 Conclusion

The process of creating Resource K-tree is the process of multi K-mean clustering according to the attributes' significance of cloud resources. QK-mean task scheduling algorithm transform the process of resource scheduling into the process of searching suitable leaf nodes in the Resource K-tree, when the task is scheduled to a leaf node of Resource K-tree, Min-min algorithm is applied to select resource server for the task. In this way, Min-min algorithm is used in a smaller scale of resources, and the efficiency of searching resource become higher. The simulation results show that QK-mean task scheduling algorithm classifies the resources according to the attribute's significance of the resource one by one, searching resources for tasks from the leaf node of the Resource K-tree, this is mean that the scale of the problem becomes smaller, so the result of clustering is more efficient.

Acknowledgment

This work is supported with the natural science funds (No. KJ2014A184 and No. KJ2013B183) from Department of Education Anhui Province, China. Many thanks to the professor Shenghui Zhao, she has put forward many constructive ideas for us. It gives us great help and encourage with her meticulously attitude to knowledge and minded guidance, which also adds to the excitement of our paper. Thanks our parents and family! Your thoughtfulness is my motivation to make great achievement. Thanks our other colleagues, they have also provided a lot of support.

References

[1] Wagstaff K, Cardie C, Rogers S, et al. 2001 Constrained K-means Clustering with Background Knowledge *Proceedings of the Eighteenth International Conference on Machine Learning* 577-84

[2] Celebi M E, Kingravi H A, Vela P A 2013 A comparative study of efficient initialization methods for the K-means clustering algorithm *Expert Systems with Applications* **40**(1) 200-10

[3] Kumar A, Kiran M, Mukherjee S, et al. 2013 Verification and Validation of MapReduce Program model for Parallel K-Means algorithm on Hadoop Cluster *International Journal of Computer Applications* **72**(8) 48-55

[4] Shannon C E 2001 A mathematical theory of communication *ACM SIGMOBILE Mobile Computing and Communications Review* **5**(1) 3-55

[5] Yao J, Dash M, Tan S T, et al. 2000 Entropy-based fuzzy clustering and fuzzy modeling *Fuzzy Sets and Systems* **113**(3) 381-8

[6] Barbará D, Li Y, Couto J. COOLCAT 2002 An entropy-based algorithm for categorical clustering *Proceedings of the eleventh international conference on Information and knowledge management ACM* 582-9

[7] Yuan F Y, Zhang X C, Luo S B 2011 Accurate property weighted K-means clustering algorithm based on information entropy *Journal of Computer Applications* **6** 065

[8] Yao J, Dash M, Tan S T, et al. 2000 Entropy-based fuzzy clustering and fuzzy modeling *Fuzzy Sets and Systems* **113**(3) 381-8

[9] Jing L, Ng M K, Huang J Z 2007 An entropy weighting K-means algorithm for subspace clustering of high-dimensional sparse data *Knowledge and Data Engineering IEEE Transactions on* **19**(8) 1026-41

[10] Rosenberg A, Hirschberg J 2007 V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure *EMNLP-CoNLL* 410-20

[11] Xu X, Xia Y, Wang Q, et al. 2014 Research about Group Decision Support System for Technology Resources Allocation of Engineering Machinery Based on Information Entropy *Journal of Software* **9**(7) 1960-8

[12] Pincus S M 1991 Approximate entropy as a measure of system complexity *Proceedings of the National Academy of Sciences* **88**(6) 2297-301

[13] Corradi A, Fanelli M, Foschini L 2014 VM consolidation: a real case based on openstack cloud *Future Generation Computer Systems* **32** 118-27

[14] Litvinski O, Gherbi A 2013 Experimental Evaluation of OpenStack Compute Scheduler *Procedia Computer Science* **19** 116-23

[15] Gupta A, Kalé L V 2013 Towards Efficient Mapping, Scheduling, and Execution of HPC Applications on Platforms in Cloud *IPDPS Workshops* 2294-7

[16] Wuhib F, Stadler R, Lindgren H 2012 Dynamic resource allocation with management objectives - Implementation for an OpenStack cloud *Network and service management (cnsm) 2012 8th international conference and 2012 workshop on systems virtualiztion management (svm) IEEE* 309-15

[17] He X S, Sun X H, Von Laszewski G 2003 QoS guided min-min heuristic for grid task scheduling *Journal of Computer Science and Technology* **18**(4) 442-51

[18] Kokilavani T, Amalarethinam DDIG 2011 Load balanced min-min algorithm for static meta-task scheduling in grid computing *International Journal of Computer Applications* **20**(2) 43-9

[19] Qian Z, Zhen L 2009 Design of grid resource management system based on divided min-min scheduling algorithm *Proc. of 2009 First International Workshop on Education Technology and Computer Science* 613-7

Authors

Guozhu Zhao, 01.08.1987, China.

Current position, grades: a researcher at Chuzhou University, China.
University studies: Master's degree in School of Information Science & Engineering from Changzhou University, China in 2011.
Scientific interest: task scheduling in cloud computing and big data.
Publications: 1 paper.

Liang Ma, 27.11.1981, Weifang city, Shandong province, China.

Current position, grades: a researcher at Chuzhou University, China.
University studies: Master's degree in Qufu Normal University of China.
Scientific interest: educational technology and computer application.
Publications: 10 papers.

Ruibin Zhao, 24.6.1983, Longnan city, Gansu province, China.

Current position, grades: a researcher at Chuzhou University, China.
University studies: Master's degree in Nanjing Normal University of China.
Scientific interest: computer graphics and high performance computing.
Publications: 10 papers.