ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RESOURCE SCHEDULING ALGORITHM BASED ON MULTI-TARGET BALANCE IN ENTERPRISE GLOUD STORAGE SYSTEM

[1, 2]YIYI XU, [2]PEIHE TANG,[2] JINWEI LIU
[1]School of Information Engineering, Wuhan University of Technology, Hubei 430070, Wuhan, China
[2]College of Computer Engineering, Guangxi University of Science and Technology, Liuzhou 545006, Guangxi,China

ABSTRACT

Resource request task is limited by scheduling length, time span, load balancing, quality of service, economic principles, safety performance and other factors in the Enterprise Gloud Storage system, it is difficult to characterize it with specific features. This paper analyzed the storage model, designed a compromised task scheduling algorithm based on multi-target balance. Algorithm's universality is verified by large-scale measurements which based on the randomly generated maps. The experimental results indicate show that in view of the computation intensity and the memory intensity load, this algorithm can perform well on aggregative indicators such as job scheduling length, scheduling cost and average energy consumption.

Keywords: *Enterprise Gloud Storage, Multi-Target Balance, Resource Scheduling*

1. INTRODUCTION

Enterprise data storage technology has undergone a remarkable transformation from the original directly storage to widely used network storage, server-attached redundant array of independent disk (RAID), centralized RAID server, network-attached storage (NAS) and storage area network (SAN), clustered storage, grid storage, cloud storage, Gloud storage[1] and many other data storage solutions have come to existence one after another.

Storage grid drawn on grid theory and technology to coordinate the combination of intelligence unit more effectively, the purpose is to construct a dynamic virtual storage under the distributed, heterogeneous, autonomous network storage resources environment, realize its internal resource sharing and resource collaboration cross the autonomy domain to provide a unified, transparent, secure access and management mechanism[2]. In the enterprise Gloud storage system, once physical server topology network is formed, it does not need to re-change. Every computer in this net topology called a node [3].

Scheduling algorithm is a resource distribution algorithm according to the strategies of system resource distribution. With the increasing of node, task scheduling in storage grid system has become extremely complex. They may be multiple application node resource requesting task within a short period of time, the amount of the required resources and urgency degree of different task are different. How to distribute resource under the circumstance of multi-node is a key problem to be solved. Traditional task scheduling algorithms often consider the unilateral attribute of task only. For example, First-Come-First-Serve (FCFS) scheduling algorithms only takes the submission time into account. In this paper, the size of storage space tasks required (the size of storage space users' data required), the urgency degree of users' expectation (the requirements of the user on the server response time) and submission time of the task are considered to designed a Multi-target based (MTB) task scheduling model to meet the needs of task scheduling of large-scale enterprise Gloud.

2. RESOURCE SCHEDULING MODEL BASED ON MULTI-TARGET BALANCE ALGORITHM

2.1 Enterprise Gloud Storage Model

The enterprise Gloud storage model is shown in Figure 1.The role of machine is basically divided into three categories: ①Resource node, this kind of nodes are the beneficiaries of resource and contributors of resources; ② Branch nodes, this kind of nodes can turn the static, passive resource nodes (local guest host) into active subject, they first extract and converse metadata), then report to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

central node. Branch nodes are usually in tree distribution, and can be expanded according to the scale of enterprise; ③Metadata service nodes, as an important assembling center, the metadata is divided into a aid-metadata (such as loading metadata, rule conversion metadata) and the driving-metadata (mapping model metadata, migration model metadata, etc.), which used for real-time loaded and views organized to the obtained metadata [1, 4]. In addition, we can set up backup server and witness server to backup and monitoring the process of extract metadata.

2.2 The Basic ideas of the Proposed Algorithm

Due to there may be multiple users submit resource request service to the metadata center service node in a short period of time, it is very important to ensure each user will be able to get the services within an acceptable wait time and service fairness. This paper establishes the resource

On the arrival of a new resource request task, it will be put to the back of the ready queue with equal (or nearly equal) priorities. At the beginning, ready queue will be scheduling by FCFS principle. If the task can be completed within the corresponding time slice, Gloud storage server will remove this task from the ready queue; otherwise, transfer it to the end of next queue and schedule obey the FCFS principle. In this way, when a long task is transferred to queue n-th, it will be scheduled in a way of time slice cycling; Only when queue 1-th to queue(i-1)-th are empty, the Gloud server will execute tasks in queue i-th .If the server is servicing a task in queue i-th, a new task is added to the queue with higher priority, then the Gloud server will put the current task to the back queue i-th and server higher-priority task.

Key of this is to determine the priority of user node with resource request task. Based on size of storage space which user required, urgency degree

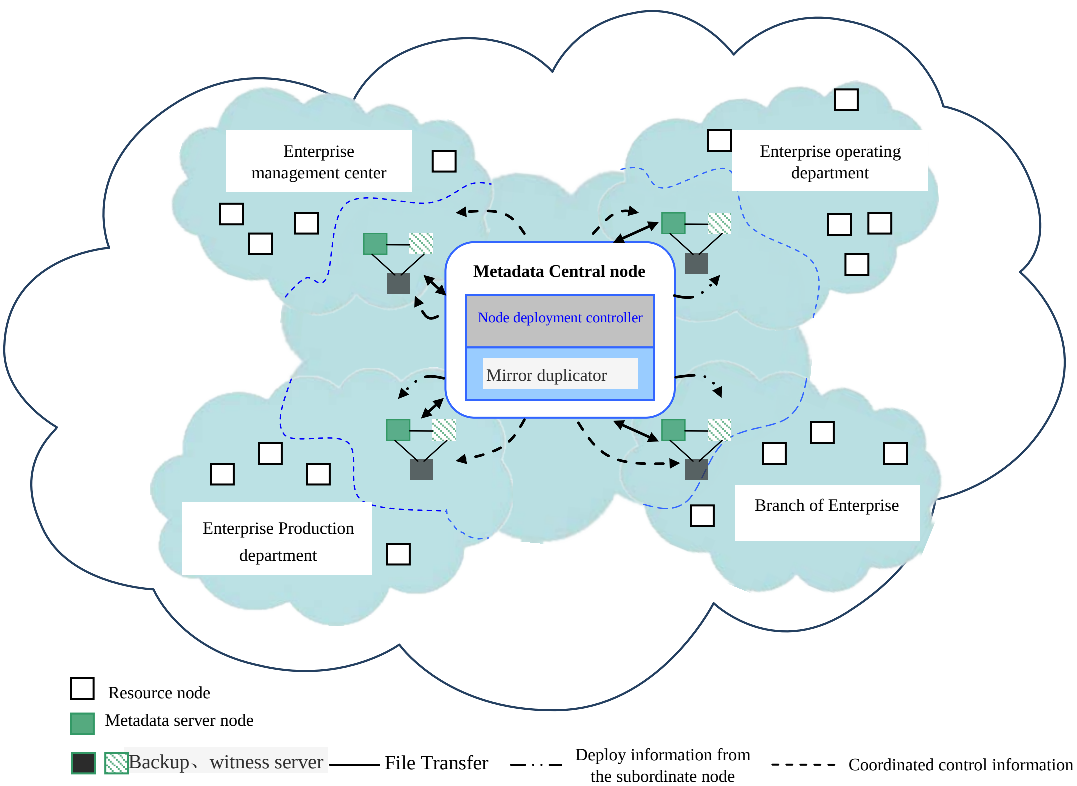

Figure 1: Typical Enterprise Gloud Storage Model

scheduling model based on multi-target balance by analytic hierarchy process. Multiple ready queues are set in this model; the priority and time slice of task execution showed in Figure 2.

of service request and the submission time of user resources request task [5], draw on the Analytic Hierarchy Process, we build a multi-target balance

Journal of Theoretical and Applied Information Technology
28th February 2013. Vol. 48 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

model to evaluate the priority of user node resource request task.

Assume the evaluation object is: x_1 (execute user 1resource requests task), x_2 (execute user 2 resource requests task), ……, x_n (execute user n resource requests task); Evaluation indexes are: f_1, f_2, $\cdots f_n$ (For example, scheduling length, time span, load balancing, quality of service, economic principles, safety performance, etc.); Index weights: $w_1, w_2, \ldots w_n$, when $0 \le w_i \le 1, \sum_{i=1}^{3} w_i = 1$, $i = 1,2,3$, then matrix A will be obtained:

$$A = \begin{array}{c} \\ x_1 \\ x_2 \\ \cdots \\ x_n \end{array} \begin{array}{c} \begin{array}{ccc} f_1 & f_2 & \cdots f_n \end{array} \\ \begin{bmatrix} a_{11} & a_{21} & \ldots a_{31} \\ a_{21} & a_{22} & \ldots a_{31} \\ \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \ldots a_{n3} \end{bmatrix} \end{array}$$

Set a_{ij} as the value of scheme x_i on indicator $f_j (i = 1,\ldots, n \;\; j = 1,2,3)$. priority function value y decreases with increasing required storage space, and increases with the increasing urgency degree of service expectation. The earlier the submission time, the larger the values of priority function. We convert the submission time into service waiting time in order to build the model, (service waiting time = scheduling time - submission time). The basic process is as follows:

First, we need to standardize the index value. The 1 to 9 scale proposed by Saaty et al. [6] is used to quantify the urgency degree of service expectation, as shown in Table 1.

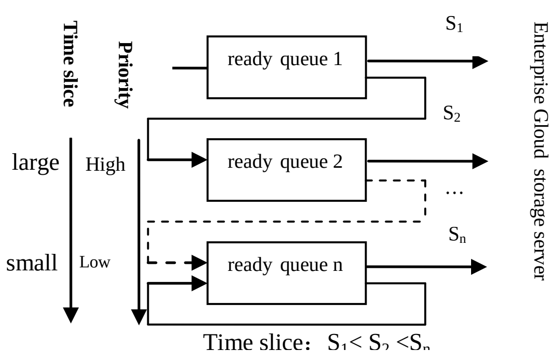

Figure 2: Multi-Target Balance Scheduling

TABLE 1: Quantification Of Qualitative Indicators

Grade	Not anxious	No hurry	General	Urgent	Very urgent
Scores	1	3	5	7	9

Because the dimensions of each impact factor are not the same, they must be standardized first before sum or linear weight sum. The Vector Normalization method is used for the standardization. For j-th indicator f_j:

$$x_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^{n} a_{ij}^2}} \qquad (i = 1,\ldots, n; \; j = 1,2,3) \tag{1}$$

After standardized, the value of X_i on the indicator f_j, equivalent to the value of resource requests Tasks to application node on indicator f_j. Then, the relative comparison method is used to determine variou index weight.

$$h_{ij} = \begin{cases} 1, & \text{When } f_i \text{ is more Important than } f_j \\ 0.5, & \text{When } f_i \text{ and } f_j \text{ are equally important} \quad (i = 1,\ldots n; \; j = 1,2,3) \\ 0, & \text{When } f_i \text{ less important than } f_j \end{cases} \tag{2}$$

Set $H = (h_{ij})_{3\times 3}$ $(i, j = 1,2,3)$,then:

$$\begin{cases} h_{ii} = 0.5 \\ h_{ij} + h_{ji} = 1 \end{cases} \qquad (i, j = 1,2,3) \tag{3}$$

The factors that effect the rational distribution and use of resources (i.e. the evaluating indicator f_1, f_2, f_n), can assigned values for f_1, f_2, f_n flexibly according to the target of system, so to determine the order of the size of f_1, f_2, f_n weight .For example, the weight value of urgency degree of user expectation f_1 is greater than the weight value of the required storage space f_2, and greater than the weight value of submission time f_3, then :

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} 0.5 & 0 & 1 \\ 1 & 0.5 & 1 \\ 0 & 0 & 0.5 \end{bmatrix} \tag{4}$$

The value of f_i can be obtained as the following weight value formula:

$$w_i = \frac{\sum_{j=1}^{3} h_{ij}}{\sum_{i=1}^{3}\left(\sum_{j=1}^{3} h_{ij}\right)} \qquad (i = 1,2,3) \tag{5}$$

Thus the qualitative weight value has converted into quantitative weight.

After we get the weigh value of each indicator and the value of arbitrary scheme on any indicator, the priority function can be obtained by the two different methods, linear weighted sum and nonlinear weighted sum:

$$y_i = w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3} \quad (i = 1,\ldots, n) \tag{6}$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$y_i = \prod_{i=1}^{n} x_{ij}^{w} (i=1,\dots,n;\ j=1,2,3) \tag{7}$$

2.3 Resource distribution Model

Due once the priority of the task is determined, the order of scheduling will be determined. Then these storage tasks will be assigned to the online resources node of the Gloud. According to the required size of storage space task and draw on the thoughts of the Buddy algorithm, The best fit algorithm is adopted to find the target resources node [7-9]. The online node distribution diagram is shown in Figure 3.

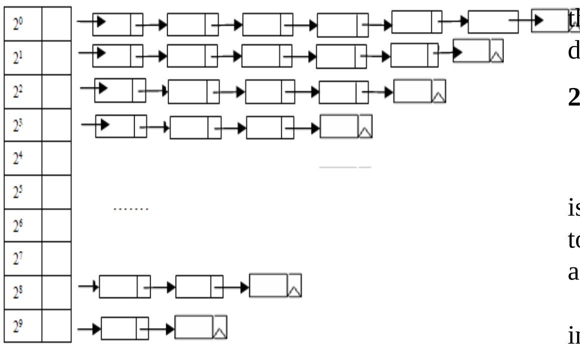

Figure 3: Online Node Distribution Diagram

The online node in Gloud storage system will be assigned to the array according to the size of available storage space [10]. Each element corresponds to a group of nodes with similar storage space size, the available storage space of the corresponding node fit formula (8).

$$\begin{cases} 2^i \le S_{group}^i < 2^{i+1}, 0 < i < 9 \\ 0 < S_{group}^i < 2^{i+1},\ i = 0 \\ S_{group}^i \ge 2^i,\ i = 9 \end{cases} \tag{8}$$

Set S_{group}^i as the available storage space of the i-th nodes group, it can be seen that the array elements increased geometrically in Figure 3. When the corresponding link-list is not empty, we can ensure the disk fragmentation rate of resource node is less than 50%.

Therefore, assuming the required storage space of storage task ST_i is NS_i, the available space of the target resource node ON_i is AS_i, disk fragmentation rate is DR_i. If the corresponding target resource node of ST_i is in the corresponding link-list, then: $2^1 \le NS_i < 2^9$. There are $k(k \in \{x \mid 1 \le x \le 8, x \in N\})$ makes $2^k \le NS_i < 2^{k+1}$, the utilization rate of ON_i is:

$$DR_i = \frac{AS_i - NS_i}{AS_i} \le \frac{AS_i - 2^k}{AS_i} < \frac{2^{k+1} - 2^k}{2^{k+1}} = \frac{1}{2} \tag{9}$$

For a given required storage space $NeedSpace\,(NeedSpace > 0)$, there is a natural number i which makes:

$$2^i \ge NeedSpace \quad \text{Or} \quad i \ge \log_2 NeedSpace \tag{10}$$

Then we obtained the range of i, round down to obtain the smallest natural number i which fits formula (10).

$$i = \lfloor \log_2 NeedSpace \rfloor \tag{11}$$

Working out $i \ge 9$ through formula (11), then find the corresponding target resource nodes to distribute.

2.4 Algorithm Description

In summary, the meta-data center services node is responsible for assigning multiple storage tasks to the corresponding resource nodes; specific algorithm is as follows:

Step 1. Define an index, determine multi-target indicators according to the needs of each application node, and then record the order and times of scheduling tasks.

Step 2. On a single assign task, allocation algorithm is calculated in accordance with the 2.3 model; If there is available space can meet the needs of the node, then the task ST_i will be assigned to the node; Otherwise, divide the storage tasks into two sub-tasks, ST_{i1}, ST_{i2}.

Step 3. Recursive complete step2, assign subtask to the corresponding resource nodes until all the requesting tasks of the user is empty.

3. ANALYSIS AND CONCLUSION

We use a Grid platform to verify the effectiveness of the proposed algorithm. Given a weighted graph G = (V, E), where vertex is constituted by server and client. The client is divided into resource nodes and application nodes, its QoS requirements are different from each other. Define length and formation rate of different data packet respectively, the data stream generated by each information source obey the Mandelbrot Zip Poisson distribution. Assumed experimental parameters are shown in Table 2, the randomly generated maps are shown in Figure 4.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The algorithm testing code was written in Java and compiled by Eclipse3.6. Assumed the number of target to be balanced is N, the worst result of time complexity is $O\left(3^{N/3}\right)$.

TABLE 2: Experimental Parameters Set

Parameter name	Parameter settings
Target area	400×400m2
Bandwidth	400Kb/s
Server	10
Resource node	50
Application node	90
Number of tasks	300
Tasks achievement rate	0.1～1.0
iteration times	300
Periodic data stream generating rate	1Packet/s

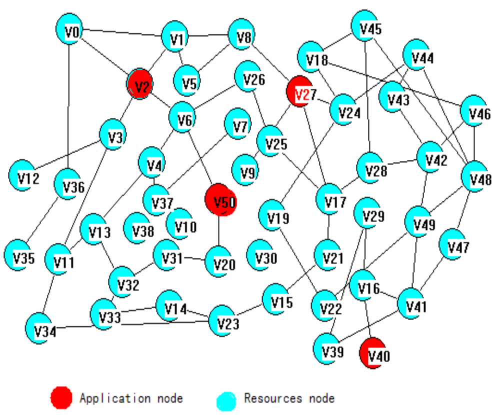

Figure 4: The Randomly Generated Map（Part）

Furthermore, under the same scale experiment conditions with the unit time, the multi-target balance based resource scheduling algorithm (MTB) has a clear advantage in resource scheduling length, average energy consumption (cost) when compared to the classical first-come-first-served algorithm (FCFS) and short-job-first algorithm (SJF), as is shown in Figure 5, Figure 6. Therefore, the proposed scheduling algorithm can effectively reduce the resource leasing costs when dealing with large-scale map-like data resources scheduling, maximize and flexibly meet the requirements of various real-time resource assign tasks.

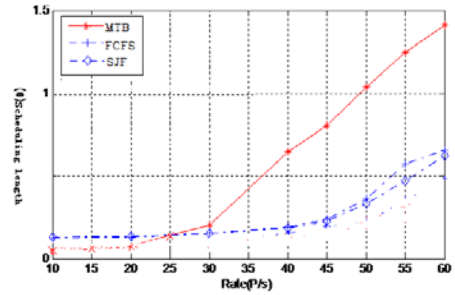

Figure5: Length Of Resource Scheduling

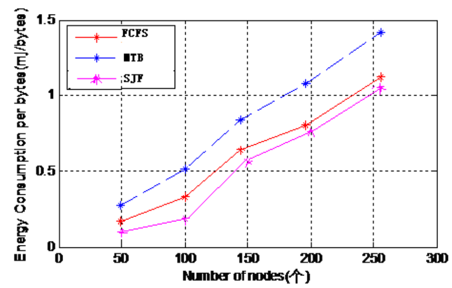

Figure6: The Average Energy Consumption (Cost)

ACKNOWLEDGEMENTS

This work was supported by the Science Research Foundation of Guangxi University of Technology (No.1261126), Natural Science Foundation of Guangxi (No. 2011GXNSFA018162) ， the Guangxi Special Major and Curriculum Integration Project (No. GXTSZY217), Guangxi features professional and curriculum integration funded construction projects (No.GXTSZY217).

REFERENCES:

[1] Tang Peihe, XU yiyi. Resource Scheduling Strategy Based on Credibility in the Enterprise Cloud Storage, *Journal of Convergence Information Technology*, Vol.7, No.16,2012, pp. 393-400.

[2] Lin WW, Qi DY, Li YJ, Wang ZY, Zhang ZL. Independent tasks scheduling on tree-based grid computing platforms, *Journal of Software*, Vol. 17, No. 11, 2006, pp. 352-2361.

[3] Tang Peihe, Liu Hao, The Research and Design of Local Area Storage Grid System, *Computer Engineering and Design*, Vol. 29, No. 16, 2008, pp. 4210-4211.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[4] Sun Hong, Zhang Xihuang. Analysis and improvement of joint routing and sleep scheduling algorithm, *Journal of Computer Applications*, Vol. 33, No. 1, 2013, pp. 115-119.

[5] LI Xin, JIA Zhi-Ping, Ju Lei et al, Energy Efficient Scheduling and Optimization for Parallel Tasks on Homogeneous Clusters, *Chinese Journal of Computers*, Vol. 35, No. 3, 2013, pp.501-602

[6] Liu Hao,Tang Peihe, Liu Jinwei, Simulation and analysis of resource request task in enterprise storage grid, *Computer Engineering and Applications*, Vol. 47, No. 19, 2011, pp. 82-84.

[7] Kenli LI，Zhao TONG, Dan LIU Etc. A PTS-PGATS based approach for data-intensive scheduling in data grids, *Computer Scinece*, Vol. 5, No. 4, 2011, pp. 513-525.

[8] LI Bo, Zhou En-wei, Shen Bin, A Simulating System for Scheduling Co-Allocation Tasks in Distributed Computing Environments, *Computer Engineering and Science*, Vol.34, No. 2, 2012, pp.82-86.

[9] LI Jian, HUANG Qingjia, LIU Yiyang, SU Sen. A Task Scheduling Algorithm for Large Graph Processing Based on Particle Swarm Optimization in Cloud Computing, *Journal of Xi'an JiaoTong University*, Vol.46, No.12, 2012, pp. 116-122.

[10] Qin X, Xie T, An Availability aware Task Scheduling Strategy for Heterogeneous Systems, *IEEE Transactions on Computers*, Vol.57, No.2, 2008, pp. 188-199.