http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 11 (1): 94-101, 2012
ISSN 1812-5638 / DOI: 10.3923/itj.2012.94.101

Cloud Simulation Resource Scheduling Algorithm Based on Multi-dimension Quality of Service

Wuqi Gao and Fengju Kang
College of Marine Engineering, Northwestern Polytechnical University, Xi'an 710032, People's Republic of China

Abstract: Aiming at the shortcomings of cloud simulation task scheduling algorithm, this paper put forward cloud simulation scheduling algorithm based on multi-dimension QoS (Quality of Service). Firstly, analytic hierarchy process in economic field was introduced into the Resource scheduling algorithm to compute every dimensional parameters weight, then the tasks was allocated to appropriate resource according to customer satisfaction, QoS distance and loading equilibrium, etc. Finally, scheduling algorithm was analyzed by theory example and was simulated with CloudSim tool package. The experiment shows that this cloud simulation scheduling algorithm not only meets customer needs for multi-dimension QoS, but shortens the simulation finishing time and greatly improves simulation resource utility rate.

Key words: Cloud computing, simulation, quality of service, resource scheduling, efficiency coefficient method, CloudSim

INTRODUCTION

In order to apply cloud computing to simulation field, scholar Li Bohu proposed the Concept of Cloud Simulation Platform in 2009 (Li *et al*., 2009). The resource service scheduling technology of the cloud simulation platform is different from other common computing and it possesses the following characteristics: (1) Multi-machine collaboration which means that cloud simulation virtual machines are distributive in geological places, but they belong to different owners logistically. (2) Dynamic isomerism which means that net virtual nodes can be dynamically added into and exited from cloud simulation system. (3) Different QoS goals (Yan-Bing *et al*., 2010; Salim *et al*., 2007; Du *et al*., 2006; Akhtar, 2007; Nehra *et al*., 2007). Cloud simulation often involves different entities, such as managers and users of cloud simulation and virtual machines. In terms of managerial mechanism, safety strategy, cost and etc, different goals are set for different entities. Some users take the shortest finishing time as priority, some take the lowest cost as priority while what the others emphasize most is whether virtual machines keep higher stability or not (Yan-Bing *et al*., 2009). Therefore, the hottest issue of present study is to achieve optimal matching between simulation tasks and virtual machines and to satisfy the applied multi-dimension QoS needs.

The traditional resource scheduling algorithms include FCFS (First Come First served) and Round-Robin algorithm. These methods can easily cause problems of task starvation, unbalanced resource and etc. (Zhongxiu, 2003). Heuristic algorithm and QoS Guided Min-Min algorithm cause resources idle and tasks extruded, not integrating into the diversity of QoS constrains. Thus, it is difficult for them to reflect dynamical heterogeneous (Li *et al*., 2008; Shi *et al*., 2011; Yan-Ping and Zeng-Zhi, 2007; Xu and Wang, 2007; Shen *et al*., 2011; Yan *et al*., 2006; Yan-Bing *et al*., 2009; Gong *et al*., 2009). A typical scheduling algorithm with the multi-dimension QoS guiding is QDDN (Wu *et al*., 2007) which is only based on the distance between the task and resource. It does not consider user's satisfaction and loading equilibrium, etc which results in unsatisfactory scheduling effect. For the above problems to be solved, Cloud Simulation Resource Scheduling algorithm based on QoS, S-CSRSA was put forward.

QUALITY OF SERVICE PARAMETER

Assuming there is no any dependent relationship among the cloud simulation tasks and the length of the submitted task is known, the m submitted tasks can be expressed as T = {t1, t2,..., tm}, n resource can be indicated as R = {r1, r2,..., rn}.

Corresponding Author: Wuqi Gao, College of Marine Engineering, Northwestern Polytechnical University, Xi'an 710032, People's Republic of China

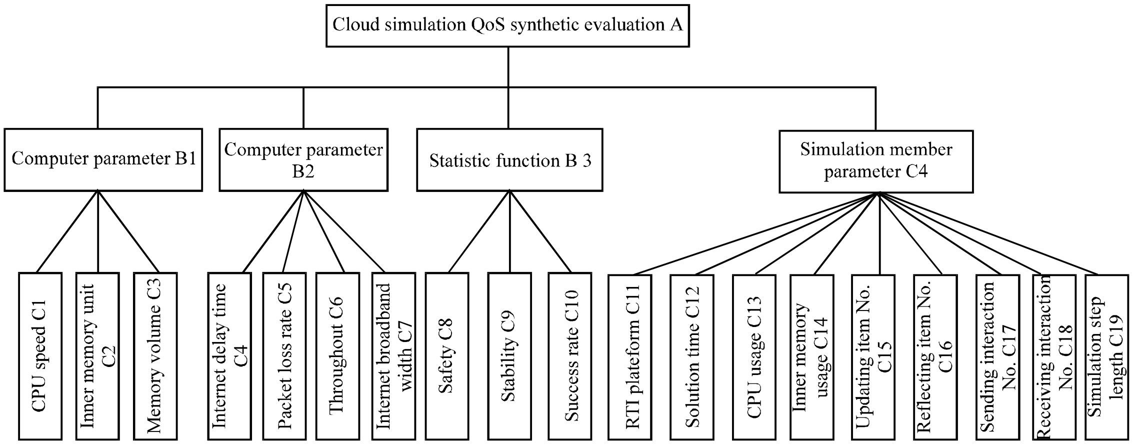

Fig. 1: Cloud simulation QoS index system

Cloud Simulation can use mass simulation resource based on the Internet, so every simulation service function need be monitored in order to select proper resource to finish simulation tasks. Also, the loading equilibrium modeling of cloud simulation platform requires a simulation member to move dynamically. if the problem of function overloading or breakdown of one simulation member arises and the loading equilibrium modeling needs to monitor simulation resource. This paper presents cloud simulation QoS index system (Fig. 1) consisting of 18 parameters in four aspects of computer, Internet, statistics and simulation member.

MULTI-DIMENSION QUALITY OF SERVICE EVALUATION

Qos Standardization:Based on k-dimension QoS capability of resource provisioning, a n*k matrix made of QoS modeling by n resources can be indicated as Qn, k:

$$Q_{n,k}=\begin{bmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,k} \\ q_{21} & q_{2,2} & \cdots & q_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ q_{n,1} & q_{n,2} & \cdots & q_{n,k} \end{bmatrix} \quad (1)$$

In the same way, based on each dimension QoS parameter's difference of simulation tasks, a m*k matrix made of QoS modeling by m tasks can be indicated as Qm, k.

Due to the big difference of each dimension parameter values for resource, tasks and for the purpose of a systemic evaluation to simulation QoS need and to resource service capability, standardizing the both matrixes is needed. This study adopts efficiency coefficient method to finish the standardization, the following are the specific rules: the bigger and better index value is defined as maximum variables; the smaller and better index value as minimum variables; the better index value at a point as stability variables; the best index value at a range as range variables.

To design efficiency coefficient method for the above four variables, respectively, Table 1 is the efficiency coefficient method formula.

Identifying user's QoS synthetic value with synthetic single efficiency coefficient method: First of all, set up tier structural modeling. Analytic hierarchy process need the problems to be categorized at the very beginning to set up hierarchy structural modeling. The hierarchy can be divided into three classes:

- The highest tier, also called objective tier, is very often to analyze the expected objectives or ideal outcomes of the problems
- The medium tier, namely principle one, includes intermediary involved in achieving the objectives which consists of a lot of tiers
- The lowest tier, measure or solution one, includes various available measures and solutions for achieving the objectives

Then, through 9 classifications judgment matrix C_{ij} can be set up to get maximum character value and eigenvector. The analytical hierarchy process takes character vector of the judgment matrix as weight vector of every index and uses summation to compute eigenvector of the said judgment matrix C. The followings are the specifics:

- Normalize every column of judgment matrix:

Table 1: Efficiency coefficient method formula

Type	Formula	Condition	Type	Formula	Condition
Max	(A-D)/(B-D)×40+60	A<B	Min	(A-C)/(B-C)×40+60	A>B
	100	A≥B		100	A≤B
Stability	[(C-A)/(C-B)]×40+60	A>B	Range	[(C-A)/(C-E)]×40+60	A>E
	[(A-D)/(B-D)]×40+60	A≤B		100	F≤A≤E
				[(A-D)/(F-D)]×40+60	A<F

A: Actual value, B: Satisfaction, C: Upper unallowable value, D: Lower unallowable value, E: Upper value, F: Lower value

$$\overline{a_{ij}} = \frac{a_{ij}}{\sum_{k=1}^{n} a_{kj}} \quad (i, j = 1, 2, \cdots, n) \tag{2}$$

- Calculate the sums of all elements in every line:

$$W_i = \frac{\overline{W_i}}{\sum_{j=1}^{n} \overline{W_j}} \quad (i, j = 1, 2, \cdots, n) \tag{3}$$

- Normalize them:

$$\overline{W_i} = \sum_{j=1}^{n} \overline{a_{ij}} \quad (i = 1, 2, \cdots, n) \tag{4}$$

Wi means relative weight vector of every element in this tier to a element in previous tier. After identifying every QoS dimensional metrics, formula 5 can be use to compute synthetic value of QoS need:

$$Uq(w) \sum s_{ij} w_j \tag{5}$$

User's satisfaction: Cloud simulation task scheduling algorithm should be used to try to meet user's QoS needs. On the condition that the bigger QoS parameter value in a dimension is, the higher the QoS is, formula (6) is adopted to compute simulation user's obtained satisfaction in a dimension. Vice versa, on the condition that the bigger the QoS parameter value is, the lower the QoS is, formula (7) is adopted. In formula (7) $j \in [1, k]$; $rQ_{i,j}$ stands for service ability provisioned by QoS parameter at Dimension J in Resource Ri, $tQ_{i,j}$ stands for need volume by QoS in Dimension J for Task I.

$$Satisfy_{i,j} = \begin{cases} \frac{rQ_{i,j}}{tQ_{i,j}}, rQ_{i,j} < tQ_{i,j} \\ 1, \quad rQ_{i,j} \ge tQ_{i,j} \end{cases} \tag{6}$$

$$Satisfy_{i,j} = \begin{cases} \frac{tQ_{i,j}}{rQ_{i,j}}, rQ_{i,j} < tQ_{i,j} \\ 1, \quad rQ_{i,j} \ge tQ_{i,j} \end{cases} \tag{7}$$

Thus, synthetic satisfaction in all operation resource dimensions for the user Task I can be defined as:

$$Satisfy_i = \frac{\sum_{j=1}^{k} satisfac_{i,j}}{k} \tag{8}$$

QoS distance measurement: QoS needs of various tasks are different, so are the QoS provisioned by various tasks. In order not to make higher QoS service capability resource occupied by lower QoS need tasks which affects the operation of other tasks and leads to increasing total simulation operating time, tasks should be allotted to its matched resource to be operated. So weight distance measurement method can be used to compute the QoS distance between simulation tasks and the resource:

$$Dis = \sqrt{\sum_{j=1}^{k} w_j (ts_{i,j} - rs_{i,j)}^2} \; w_j \ge 0, j=1,2,\ldots,k, \sum_{j=1}^{k} w_j^2 = 1 \tag{9}$$

S-CSRSA ALGORITHM

Based on multi-dimensional QoS cloud simulation resource scheduling algorithm, strategy of this algorithm is: to arrange the order and allot virtual machines according to synthetic efficiency coefficient method of tasks. At first, select virtual machines with higher satisfaction to allot. If there is only one virtual machine that obtains the highest satisfaction, allot the task to it; if there are many of them with the highest satisfaction, select unused virtual machines to achieve loading equilibrium; if there is no unused virtual machines, allot the task to the virtual machine with shortest distance among the virtual machines and the task. The specific flowchart is showed in the following Fig. 2:

The algorithm of flow chart is as:

Step 1: Set up Matrix $Q_{m,k}$. with all task QoS needs of the to be scheduled task set T

Step 2: Set up Matrix $Q_{n,k}$ with QoS capabilities of all virtual machines in available resource set R

Step 3: Emerge $Q_{m,k}$ and $Q_{n,k}$ to set up a new Matrix $Q_{m+n,k}$.

Step 4: Get $S_{m+n,k}$ by standardising the Matrix $Q_{m+n,k}$ by adopting of the Table 1 method

Step 5: Isolate $S_{m+n,k}$ to get the standardised task QoS Matrix $ST_{m,k}$ and the virtual machine QoS Matrix $SR_{n,k}$

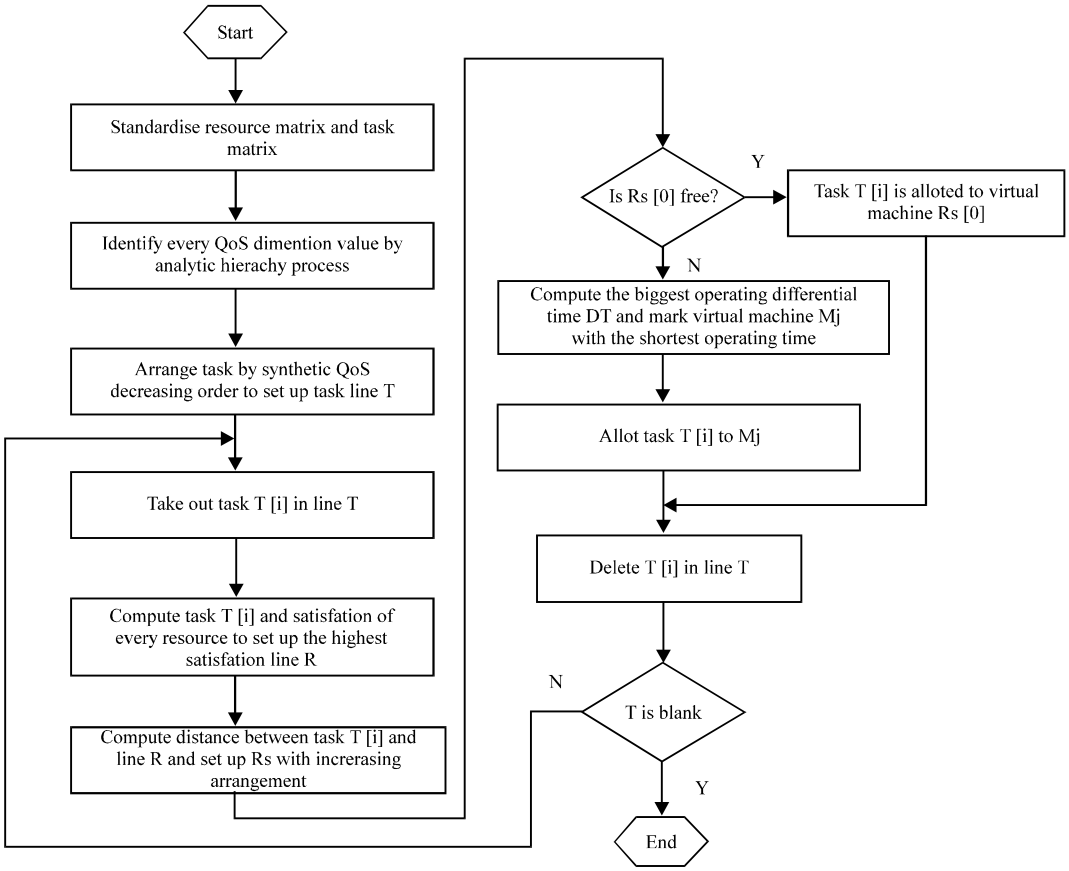

Fig. 2: S-CSRSA algorithm flowchart

Step 6: Use analytic hierarchy process to compute the QoS weights

Step 7: Compute the synthetic QoS need volume of every task, arrange in the big-to-small order in accordance with the task synthetic QoS need volume and allot firstly the bigger synthetic QoS need volume to operating lines of the proper virtual machines

Step 8: Take out the first task t0 in the task lines, compute the satisfaction of this task at every resource, then find out the resource which gives highest satisfaction to this task and store it into resource line R

Step 9: Compute the distance between Task T [i] and Line R and form Rs with the increasing arrangement

Step 10: If Rs [0] is unused, allot the task T [i] to virtual machine Rs[0] and turn to Step 13.

Step 11: Compute the biggest differential time DT of operating task of all virtual machines in Line R with the highest satisfaction, mark the virtual machine Mj getting the shortest operating time

Step 12: If the operating time DT is longer than operating time of T[i], allot T[i] to virtual machine Mj

Step 13: Delete T[i] from T

Step 14: Decide whether T is blank, turn to Step 8 if not

Step 15: Finish

S-CSRSA algorithm theory test: Given that at present there are 3 cloud simulation tasks, t1, t2, t3, their QoS need dimensional value is 4, they are CPU operation speed, Internet broadband width, stability and simulation member task length, respectively. Of them, the first three elements are positive vector, the last one negative vector. Task dimensional QoS need and the available QoS index value of virtual machines are indicated in the following Table 2 For easy comparison, the index value is the relative value from 0 to 1 on average.

Get the standardized virtual machine Matrix $vS_{3,4}$ and task Matrix $cS_{3,4}$ by using Table 1 to standardize the positive and negative vectors, respectively:

Table 2: Task dimension QoS need and the available index value of virtual machine

Content	CPU operation speed	Internet broadband width	Stability	Simulation member task length
Cloudlet 1	0.8	0.7	0.7	0.6
Cloudlet 2	0.8	0.9	0.8	0.3
Cloudlet 3	0.5	0.8	0.8	0.5
VM1	0.8	0.6	0.5	0.7
VM2	0.8	1.0	0.9	0.5
VM3	0.9	0.8	0.8	0.4
VM4	0.8	0.9	0.8	0.4
VM5	0.7	0.2	0.4	0.9

$$vS_{3,4}=\begin{bmatrix}0.75 & 0.50 & 0.20 & 0.33\\ 0.75 & 1.00 & 1.00 & 0.67\\ 1.00 & 0.75 & 0.80 & 0.83\\ 0.75 & 0.87 & 0.80 & 0.83\\ 0.50 & 0.00 & 0.00 & 0.00\end{bmatrix}$$

$$cS_{3,4}=\begin{bmatrix}0.75 & 0.63 & 0.60 & 0.50\\ 0.75 & 0.88 & 0.80 & 1.00\\ 0.00 & 0.75 & 0.80 & 0.67\end{bmatrix}$$

By the use of analytical hierarchy process, every dimensional QoS resource weight value can be computed, they are w1 = 0.123, w2 = 0.292, w3 = 0.319 and w4 = 0.266, respectively. By the use of formula (5) the total QoS need of every task can be computed, respectively they are u1 = 0.601, u2 = 0.870, u3 = 0.652. Therefore, the synthetic QoS of Task T2 is the biggest, Task T2 is firstly allotted to the operation line of the resource waiting for being operated, Task T1 is the last. By the use of formula (8), the satisfactions of virtual machines from v1 to v5 in t2 can be computed, respectively they are 0.68, 0.90, 0.91, 0.94, 0.48 and virtual machine v4 has task t2 get the highest satisfaction. Thus, task t2 is allotted to the operation line to be operated.

By formula (8) the satisfactions of virtual machines from v1 to v5 in t3 can be computed, respectively they are 0.77, 1.0, 1.0, 1.0, 0.58 and the virtual machine v2, v3, v4 have task t3 get the highest satisfaction and make themselves marked with the highest satisfaction for task t3. Thus, task t3 is allotted to the virtual machines with the shortest distance between it and the virtual machines v2, v3, v4 to be operated. By formula (9) the distances between Task t3 and v2, v3, v4 can be computed, respectively they are 0.31, 0.36 and 0.28. The distance between Task t3 and the virtual machine v4 is the smallest, however, the virtual machine v4 has got Task t2 at this time, the distance between Task t3 and the virtual machine v2 is the second smallest and v2 is free, so allotting Task t3 to v2 to be operated achieves loading equilibrium among virtual machines.

By formula (8) the satisfactions of virtual machines from r1 to r5 in t1 can be computed, respectively they are 0.86, 1.0, 1.0, 1.0, 0.60, the virtual machine v2, v3, v4 have task t1 get the highest satisfaction and make themselves marked with the highest satisfaction for task t1. Thus, task t1 is allotted to the virtual machines with the shortest distance between it and the virtual machines v2, v3, v4 to be operated. By formula (9) the distances between Task t1 and v2, v3, v4 can be computed, respectively they are 0.31, 0.23 and 0.25. The distance between Task t1 and the virtual machine v1 is the smallest and v3 is free at the time, so allotting Task t1 to v3 to be operated.

SIMULATION RESULT AND ANALYSIS

CloudSim (Calheiros *et al*., 2009, 2010) is a cloud computing simulator device which is developed under the team led by professor Rajkumar Buyya from Melbourn University, Australia. In it, the method of bind Cloudlet To VM (int cloudletID, int VMID) provisioned by Data center Broker binds every single task to single fixed VM (virtual machine) to be operated which materializes the resource research need for the reasonable and special-tasked VMs. The type of Data center Broker of expansion CloudSim platform is to self-define bind Cloudlet To VMO Method to achieve its own modeling strategy. After the expansion, the CloudSim platform need to be retranslated and set up.

At first, add relative QoS index in coudlet and VM of CloudSim. For easy testing, this paper chooses four parameters of CPU operation speed, Internet broadband width, stability and simulation member task length as example to test.

Through simulation platform this section is mainly to model task finishing time, resource utility rate, loading equilibrium, user satisfaction and etc of algorithm of QDNN and S-CSRSA proposed by Wu *et al*. (2007). Net environment of 10 VMs and only one PE for every resource were designed, the processing capability of every resource is 515 (MIPS). Resource description is indicated in Table 3.

Computing time comparison: Figure 3 and 4 are the computing time comparison of QDNN and S-CSRSA algorithms. From the aspects of operating result, for the same task and same virtual environment, computing time of QDNN algorithm is 751.63 while that of S-CSRSA algorithm is 377.57, the latter saves time by a half of the former.

Comparison of satisfaction and loading equilibrium: Establish the following regulations to the scheduling VMs

```
==== OUTPUT =======
Cloudlet ID     STATUS     Data   Center ID     VM ID     Time     Start time     Finish time
    0              2                  0         67.01              0              67.01
    4              2                  0         57.97              67.01          124.98
    2              2                  2         144.68             0              144.68
    3              2                  3         149.28             0              149.28
    8              2                  0         106.44             124.98         231.42
    6              2                  2         96.04              144.68         240.72
    7              2                  3         106.52             149.28         255.8
    1              2                  1         377.54             0              377.54
    5              2                  1         139.01             377.54         516.55
    9              2                  1         235.08             516.55         751.63
***** Power data center: Data center_0*****
User id          Debt
3                499.6
*************************
QDDN        finished
```

Fig. 3: Simulation Result of QDNN Algorithm

```
==== OUTPUT =======
Cloudlet ID     STATUS     Data   Center ID     VM ID     Time     Start time     Finish time
    9              2                  3         104.79             0              104.79
    7              2                  0         109.07             0              109.07
    8              2                  3         103.91             104.79         208.7
    3              2                  2         211.38             0              211.38
    2              2                  0         104.68             109.07         213.75
    6              2                  3         67.84              208.7          276.54
    0              2                  0         67.12              213.75         280.87
    5              2                  2         87.88              211.38         299.25
    4              2                  3         56.72              276.54         333.26
    1              2                  1         377.57             0              377.57
***** Power data center: Data center_0*****
User id          Debt
3                499.6
*************************
S-QDDN        finished
```

Fig. 4: Simulation result of S-CSRSA algorithm

Table 3: VM QoS parameters

VM No.	CPU operation speed	Internet broadband width	Stability	Simulation member task length
VM1	0.7	0.6	0.3	0.6
VM2	0.1	0.1	0.9	0.5
VM3	0.7	0.7	0.3	0.6
VM4	0.9	0.9	0.2	0.6
VM5	0.7	0.6	0.3	0.5
VM6	1.0	1.0	0.2	0.6
VM7	0.5	0.4	0.6	0.5
VM8	0.1	0.1	1.0	0.5
VM9	0.5	0.5	0.6	0.5
VM10	1.0	1.0	0.1	0.6

and user tasks: All QoS value of user tasks should use rand() to get a random number ranged 0-1. Look at Table 3 for VM QoS parameters.

Table 4 is the simulation result of algorithms of QDDN and S-CSRSA. The above experiment result shows that user satisfaction apparently improves with the algorithm S-CSRSA than that of QDDN. This is because, when allotting tasks, algorithm of QDDN only takes distance difference between user tasks and resource into account rather than other elements. According to S-CSRSA algorithm, VMs can get the highest satisfaction to user tasks. In order not to make QoS required tasks occupy resources with the higher QoS capability, S-CSRSA algorithm allots user tasks to VMs with smallest distance to itself and with the highest satisfaction. When computing distance, S-CSRSA algorithm fully considers every dimension's service capability difference of the resource and gives different metrics to different parameters to show the better match between user tasks and VMs.

Figure 5 shows that VM utility rate of S-CSRSA algorithm keeps around 54% on average while that of QDDN algorithm only remains around 32% on average, because a lot of resources in net blocks can make a task get the highest satisfaction at the same time. When resource is unused, S-CSRSA algorithm moves the Not

Table 4: Comparison of task user satisfaction

Task No.	QDDN	S-CSRSA	Improved
10	0.845	0.97	14.793
20	0.823	0.963	0.9695
30	0.844	0.9695	14.870
40	0.8385	0.696	15.564
50	0.854	0.972	13.871
60	0.8635	0.973	12.681
80	0.8595	0.973	13.205
100	0.8705	0.9745	11.947

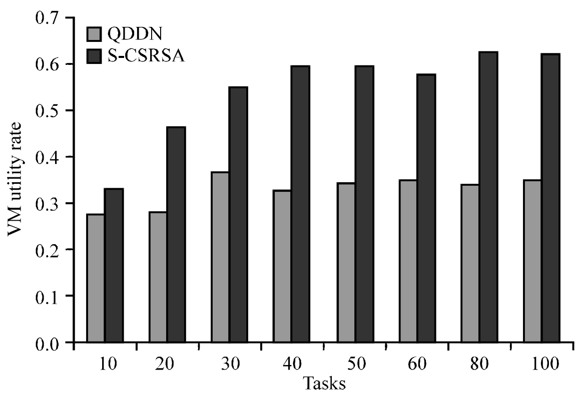

Fig. 5: VM Utility Rate Comparison between S-CSRSA and QDDN

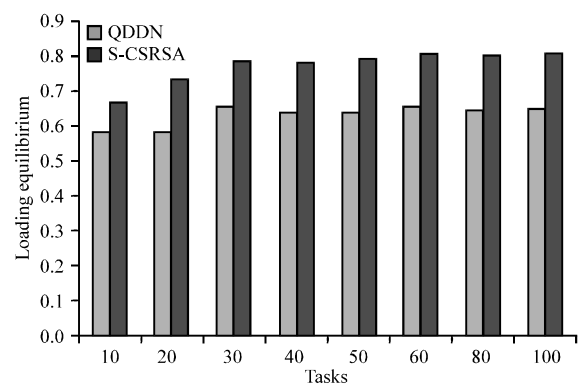

Fig. 6: Loading Equilibrium Comparison between S-CSRSA and QDDN

implemented tasks to VMs with the guaranteed satisfaction to be operated. While QDDN algorithm does not take this into account, so its resource utility rate is obviously lower than that of S-CSRSA algorithm. Due to the said reason, the total task finishing time of S-CSRSA algorithm is shorter than that of QDDN algorithm.

With the increase of tasks numbers, the difference of total task finishing time is bigger and bigger. The resource loading equilibrium from the both algorithms is indicated in Fig. 6. When allotting tasks by S-CSRSA algorithm, resource capability is fully used and operating difference of various resources is small, therefore, loading equilibrium of S-CSRSA is clearly higher than that of QDDN. When allotting tasks by QDDN algorithm, it causes many tasks allotted to a single resource to be operated, so its loading equilibrium is lower.

CONCLUSION

This study proposes a cloud simulation task scheduling model supporting QoS and introduces S-CSRSA algorithm in details on the bases of this model. Through algorithm example and cloud simulation platform testing based on ClougSim, S-CSRSA algorithm guarantees satisfaction of user tasks and good performance in resource utility rate and loading equilibrium.

REFERENCES

Akhtar, Z., 2007. Genetic load and time prediction technique for dynamic load balancing in grid computing. Inform. Technol. J., 6: 978-986.

Calheiros, R.N., R. Ranjan, C.A.F. De Rose and R. Buyya, 2009. CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services. http://arxiv.org/ftp/arxiv/papers/0903/0903.2525.pdf

Calheiros, R.N., R. Ranjan, A. Beloglazov, C.A.F. De Rose and R. Buyya, 2010. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. http://www.buyya.com/papers/CloudSim2010.pdf

Du, Y., C. Jiang and Y. Guo, 2006. Towards a formal model for grid architecture via petri nets. Inform. Technol. J., 5: 833-841.

Gong, H., J. Yu, Y. Hou and H. Liu, 2009. User QoS and system index guided task scheduling in computing grid. Comput. Eng., 35: 52-54.

Li, B., X. Chai, B. Hou, T. Li and Y. Zhang, 2009. Networked modeling and simulation platform based on concept of cloud computing: Cloud simulation platform. J. Syst. Simulat., 21: 5292-5299.

Li, J., X. Lu and S. Dong, 2008. Research on resource scheduling strategies for gird based on grid sim. Comput. Sci., 35: 95-97.

Nehra, N., R.B. Patel and V.K. Bhat, 2007. Load balancing with fault tolerance and optimal resource utilization in grid computing. Inform. Technol. J., 6: 784-797.

Salim, M., A. Manzoor and K. Rashid, 2007. A novel ANN-based load balancing technique for heterogeneous environment. Inform. Technol. J., 6: 1005-1012.

Shen, H., Z. Ding and H. Chen, 2011. Reliable web services selection based on finite state machine model. Inform. Technol. J., 10: 1662-1672.

Shi, Y.J., G.J. Shen and W. Chen, 2011. Solving project scheduling problems using estimation of distribution algorithm with local simplex search. Inform. Technol. J., 10: 1374-1380.

Wu, Z., J. Luo, F. Dong and X. Ni, 2007. QoS deviation distance based negotiation algorithm in grid resource advance reservation. Proceedings of the 11th International Conference on Computer Supported Cooperative Work in Design, April 26-28, Melbourne, Vic., pp: 605-610.

Xu, Y.Q. and Z.X. Wang, 2007. Optimized grid schedule based Qos guided min-min. Comput. Appl., 27: 215-216.

Yan, T., W. Li and C.Y. Li, 2006. QoS guided task scheduling heuristic in computational grid environments. Microelectron. Comput., 23: 107-110.

Yan-Bing, L., C.H. Jie and X. Shi-Yong, 2009. Grid task scheduling algorithm based on QoS similarity. J. Chongqing Univ., 3: 416-420.

Yan-Bing, L., S. Ming-Sheng and X. Yun-Peng, 2010. Grid Resource Management Technology and High Performance Scheduling. Science Press, Australia.

Yan-Ping, C. and L. Zeng-Zhi, 2007. E-WsFrame: A framework support QoS driven web services composition. Inform. Technol. J., 6: 390-395.

Zhongxiu, S., 2003. Operating System Tutorial. High Education Press, Beijing, China.