Resource Scheduling Algorithm Applied in Cloud Computing

Hean LIU[1], Shuguang WU[2]
1.Hunan City University, Yiyang, 413000, China
2.Modern Education Technology Center, Hunan University of Technology, Zhuzhou, 411201, China

Abstract: To improve the efficiency and accuracy to discovery resource in cloud system, this paper proposes CICP resource scheduling algorithm. The algorithm is proposed based on cloud resource scheduling modeling, cloud applications preference measurement and multidimensional objective optimization function. According to application preference information, the algorithm makes antibody immune clone operation, immune genetic operation and immune selection operation on antibody priority allocation preferences, and then the paper gives specific steps of the algorithm to conduct experimental validation on availability, load balancing and efficient time. Experimental results show that the proposed algorithm can satisfy the requirements of resource scheduling in cloud computing.

Keywords: Preferences measure; Utility optimization; Immune genes; Clone

1. Introduction

As a new large-scale distributed computing paradigm derived by business, Cloud computing utilizes key technologies like abstraction, virtualization, instantaneous deployment and broadband network to unify the form of service via internet resource interconnection, interoperability and interoperability. Browser used multi-terminal, multi-platform and multi-network as standard to offer on-demand, dynamic configuration, elastic expansion, low price, high availability and high reliability nontrivial QOS (quality of service) services including computing, storage, platform, etc [1-5]. At any time and any place. From a theoretical perspective, cloud computing provide low-cost, high efficiency, high availability of services through network data centers and dynamic resource pools, which effectively achieve the goal of utility computing. In practice, users can access to the services they need via the Internet with a standard browser at any time and any place, and the resources like storage space, computing power are application platform no longer limited [6-9]. All of these offer efficient solutions for the sharp increase in hardware costs, computing and storage capacity, computing services and new requirements brought by the development of Web3.0, which achieves widely attention from the government, enterprises and research institutions. In cloud systems storage, clusters, data, software, network, etc. constitute to cloud resources, which is the underlying support environment. For cloud resource scheduling features, discovery and scheduling mechanism of researching efficient, economical, rapid resource are the main issue of this paper. Resource scheduling is a core and challenging important component of cloud systems. Compared with traditional distributed computing paradigm, cloud computing has the following major differences: 1) flexible network access mechanisms. Users can take advantage of a variety of flexible heterogeneous terminals to obtain instantaneous efficient service via the Internet; 2) Quantifiable service. Cloud system provides users with quantifiable services to make real-time service status monitoring and optimization; 3) Rapid supply of services. Cloud system through virtualization and other related key technologies provides users with various services like instant deployment and dynamic configuration; 4) Services according to need. Cloud system conducts resources configuration and reconfiguration based on user’s pay; the cost factor is another important factor of resource scheduling; 5) Dynamic resource pools. Cloud system based on different levels of virtualization technologies, shields resources distributed and heterogeneous characteristics in cloud system to provide users with transparent and unified service interface, so that user's service requests can rapid completed [10-15]. According to the special requirements of the system, a comprehensive, in-depth, systematic analysis and research are conducted. On one hand, the characteristics of application are applied to efficiently reduce the search space resources and improve resource discovery efficiency and accuracy, so that Quality of customer service can be effectively ensured. On the other hand, for the user's status changes, resources selection and redirection are dynamically conducted. According to the user's application preferences, the user's economic QOS efficient can be effectively met which can effectively improve user satisfaction.

2. Modeling

2.1. Cloud resource scheduling model

Priori preferences are divided into rigid preferences and the elasticity preferences: Rigid preferences mean that the application preferences must be completed within the specified range, otherwise the mission will fail. Elasticity preferences mean such preferences are completed within the zone arranged by user o_j, also they can be completed outside (at this point the users' utility are low).

When the k-dimensional feature attribute $e_i(k)$, 0≤k≤t of resource e_i meet the rigid preferences, the preference interval of user o_j is $[e_i(k)low, e_i(k)up]$. If the preference actual value $e_i(k)$ real of the k-dimensional feature attribute is low, the preference value $we(k)$ of the k-dimensional feature attribute meets equation (4); otherwise satisfies the formula(1).

$$weij(k)=\begin{cases}\dfrac{e_i(k)_{up}-e_i(k)_{real}}{e_i(k)_{up}-e_i(k)_{low}}, e_i(k)_{low}\le e_i(k)_{real}\le e_i(k)_{up} \\ 0, \quad otherwise\end{cases} \tag{1}$$

Then, the preference weight $q(k)$ with k-dimensional feature attribute of the users satisfies the equation (8). The preference vector of user $u(j)$ regarding resource e_i is as following:

$$q_{ij}=(q_{ij}(0),q_{ij}(1),...qi_{ij}(t)),\sum_{i=0}^{t}q_{ij}(k)=1 \tag{2}$$

2.2. Objective optimization function of multidimensional QOS

The t-dimensional QOS attribute vector $qi_j=\left(qij(1),qij(k),...qij(k)\right)$ of user $u(j)$ about the resource e_i conduct t-QOS space. Wherein, $qi_j(k)$, 1≤k≤t is the QOS of k characteristics attributes of the user $u(j)$ on resource e_i. User utility of k characteristics attributes of the user $u(j)$ on resource e_i is $Uti_{sys}qi_j(k)$, $1\prec k\prec t$. Thus, in t -QOS space, the cloud resource scheduling multiple QOS objective optimization utility function is as formula(3)

$$\begin{cases}Of(Uti_{sys})=\max\sum_{k=0}^{t}\sum_{j=0}^{m}\sum_{i=0}^{n}w_{ij}(k)\bullet uti_{ij}(qi_j(k)) \\ s\square t, \quad 0\le w_{ij}(k)\le 1,\sum_{i=0}^{t}w_{ij}(k)=1\end{cases} \tag{3}$$

3. Immune Clone Preferences and Multidimensional Cloud Resource Scheduling Algorithm

Immunity refers to the identification of "Self" and "Non-self", and it will discharge the functions of non-self. The main actions of immune clone include immune cloning, genetic manipulation and cloning immune selection operation.

3.1. Immune clone

With a fixed length string integer number of tasks, chromosome of antibodies (Task resource scheduling scheme) are generated. Wherein, each gene locus represents the allocated resources number. The 0 Generation of the i_{th} antibody cA_i in set U's tasks set satisfies the equation (4), while the 0 generation antibody group with the size of N_{Ab} satisfies the equation (5).

$$cA_i(0)=(ca_i^1,ca_i^2,...ca_i^{mt}),ca_i^k\in\{1,2,...,n\} \tag{4}$$

$$cA(0)=(cA_1(0),...,cA_i(0),...,cA_{N_{Ab}}(0)),i\in\{1,...,N_{Ab}\} \tag{5}$$

Wherein, c is the resources number of the k_{th} task te_i of the i_{th} antibody. re is the tasks number of the user set, and satisfies the formula (6). Since Min-min algorithm has better load balance, initial antibody is generate by using the Min-min algorithm.

$$re=\sum_{j-0}^{r}te(i),i\in\{1,2,...,r\} \tag{6}$$

Wherein, te_i is the number of cloud tasks of the j_{th} user.

Between the antibody cA_i and the antigen $AgAb_i$ satisfies formula, which reflects the level of antigen. If the antigen $AgAb_i$ is big, the total utility of antibody cA_i is large.

Affinity $AgAb_i$ between antibody cA_i and other antibodies satisfies the formula. If the affinity $AgAb_i$ is small, the similarity between antibodies is great and the inhibition between antibodies is strong. When $AgAb_{i=0}$, the two antibodies are identical.

Wherein, $\|CAi-CAj\|$ is the Euclidean distance between antibody cA_i and antibody cA_j. N_{Ab} is the clone operation T_c^C of antibody population CA, which can be seen in formula (7).

$$CA'(it)=T_c^C(CA(it))=\left[T_c^C(CA_1(it)),T_c^C(CA_2(it)),T_c^C(CA_{N_{Ab}}(it))\right]^T \tag{7}$$

3.2. Immune genetic operation

Immune genetic operation is mainly including clones recombinant and clone variation. This paper chooses the better performance with multi-objective optimization $Sim-ulated$ Binary Crossover (SBX) and Polynomial Mutation (PM) [13]. SBX restructuring is show as formula.

$$ca'^{k}_{i}=\begin{cases}0.5\square\left[(1+\beta_k)\square ca^{k}_{i}+(1-\beta_k)\square ca^{k}_{i}\right],\\ if r(0,1)\geq 1/\alpha_k;\forall i,j,i\neq j;k\in\{1,2,...,mt\},\\ 0.5\square\left[(1-\beta_k)\square ca^{k}_{i}+(1-\beta_k)\square ca^{k}_{j}\right]\\ otherwise;\forall i,j,,i\neq j;k\in\{1,2,...,mt\},\end{cases}\qquad(8)$$

Wherein, β_k satisfies the formula, α_k satisfies the formula; $r(0,1)$ is the random Gaussian distribution number within (0,1); ca_j^k is results after the reorganization with ca_j^k and ca_j^k.

$$\beta_k=\begin{cases}\left[\alpha_k\square u\right]\dfrac{1}{\eta_c+1}, if\ u(0,1)\leq 1/\alpha_k\\ \left[2-\alpha_k\square u\right]-\dfrac{1}{\eta_c+1}, otherwise\end{cases}\qquad(9)$$

Wherein, u,i is the random Gaussian distribution number within (0,1); η is a recombinant index parameter; α_k satisfies the formula.

Wherein, ca_u^k, and ca_I^k represents the minimum and maximum of k genes position of the whole antibodies; the greater the η is, the more similar the reorganization and new antibodies are.

PM variation is defined in formula (10).

$$ca'^{k}_{i}=ca^{k}_{i}+\delta_k\square(ca^{k}_{u}-ca^{k}_{l}),i\in\{1,2,...,N'_{Ab}\}\qquad(10)$$

Wherein, ca_I^k and $ca_i'^k$ are the values of the k gene locus before mutation and after mutation. σk is tuning parameters of PM variation, and satisfies the formula (11).

$$\delta_k=\begin{cases}\left[\begin{array}{c}2\bullet v+(1-2\bullet v)\square\\ \left[\dfrac{\max((ca^{k}_{u}-ca^{k}_{i}),(ca^{k}_{i}-ca^{k}_{l})}{ca^{k}_{u}-ca^{k}_{l}}\right]^{\eta_m+1}\end{array}\right]^{\frac{1}{\eta_m+1}}, if v(0,1)\leq\dfrac{1}{2}\\ 1-\left[\begin{array}{c}2\square(1-ca^{k}_{i})+2\square(1-ca^{k}_{i}-0.5)\square\\ \left[\dfrac{\max((1-ca^{k}_{u}),(ca^{k}_{i}-2\square(1-ca^{k}_{l}))}{ca^{k}_{u}-ca^{k}_{l}}\right]^{\eta_m+1}\end{array}\right]^{\frac{1}{\eta_m+1}}, otherwise\end{cases}\qquad(11)$$

PM mutation probability pm is defined in formula (12).

$$p_m=\begin{cases}(1+p)\square P_{\min}-2\square p\square P_{\min}\square\left[\dfrac{it}{it_{\max}}\right],\\ if\quad it\leq it_{\max}/2;p\in(0,1),\\ P_{\min}, otherwise\end{cases}\qquad(12)$$

Wherein, $p_{\min}$ is the predefined minimum mutation probability; p is the predefined variation value; it is the current cloning algebra; $it_{\max}$ is the predefined maximum algebra.

SBX restructuring operation achieves the exchange of information between the antibodies. They have adaptive convergence along with the evolution of the antibody population; PM mutation makes tuning adaptive to the antibodies; pm achieve the PM variation by selecting antibodies with appropriate probability in different generation of cloning process.

3.3. Immune selection operation

Clone selection operation selects the optimal antibody of fitness c from the offspring antibody population of clone operation and immune gene operation; thereby a new population is formed. Cloning operation t_c^s of antibody population CA″ is shown in the formula (13).

$$CA'''(it+1)=t^{c}_{s}(CA''(it))=t^{c}_{s}(CA''_1(it),CA''_2(it),...,CA''''_{p+N'_{Ab}}(it))^T\qquad(13)$$

$$=(CA'''_1(it),CA'''_2(it),...,CA'''_{N_{Ab}}(it)^T$$

Recombination and mutation of high frequency easily lead to the loss of dominant mode, so in order to preserve the favored model in offspring antibody genes, the algorithm is designed in accordance with the size of the fitness $AgAb_i$, and select the same number of offspring individuals with the parent personal to make up the next generation of antibody population, namely

$$CA(it+1)=T^{c}_{s}(CA(it)+CA''(it))\qquad(14)$$

3.4. Algorithm description

On the basis of diverse antibody symbiosis in immune response and the fact of a small number of active antibodies, cloud resource scheduling multiple QOS optimization with the application preferences is combined. By using immune clone operation, immune genetic manipulation and cloning elective operation, cloud resource scheduling by immune clonal with preference (CICP) optimization algorithm is proposed as shown in algorithm 1.

Algorithm 1 CICP algorithm

Input: cloud resource set R, cloud user set U, cloud probability matrix

$p_{m\times n}$, preference interval, 5-QOS utility interval, minimal mutation probability $p_{\min}$, magnitude mutation p, and maximum generations $it_{\max}$;

Time complexity of the CICP algorithm consists of immune clone, immune genes and clone selection. Let M is the number of the objective function; NAb is for scale of the antibody group CA (0); antibody scale after completing the immune cloning operation is as N_{Ab}; so the antibody scale after immune genetic operation is as followings:

$$N'_{AB}(1+p)=N'_{AB}+N'_{AB}/3\rfloor, N_c>N_{Ab}\qquad(15)$$

This is the related settings related to the size of the clone and generally $Nc=3$ ·Nab is taken. In the generation of the cloning, the time complexity of the target value in initial population and is calculated as $o\left(M\times N_{Ab}\right)$; the worst time complexity of immune clone operation is $o\left(M\times N_c\times\log\left(N_c\right)\right)$; the worst time complexity of immune genetic clone operation is $o(M\times N'_{Ab}.(1+p))$; the time complexity clone selection operation is $o\left(M\times\log\left(N'_{Ab}\right)+M\times N'_{Ab}\right)$.

4. Experimental Simulation and Analysis

4.1. Experimental settings

CloudSim-based cloud simulation computing platform establishes the resource scheduling environment with 10 data centers and 600 user; its main parameters are shown in Table 1. Virtual machine is 100, of which VCPU number of each virtual machine are between [1, 2] ; the number of tasks for each user are between [5,10]; the amount of calculation is between [10000,20000] MI; the data traffic is between [10,30] Mbps. Each task submits to Poisson distribution and mutually independent. The final completion time dt_j of task te_i satisfies formula (16).

Table 1. Parameter Setting in Data Center

Datacenter ID	Machine number	PE/M achine	Cost	Architecture and OS
DC_0.DC_1	2	1~4	3	X86/Linux
DC_2.DC_3	4	1~4	6	X86/Linux
DC_4.DC_5	6	4~8	9	X86/Solaris
DC_6.DC_7	8	4~16	12	X86/Solaris
DC_8.DC_9	10	8~16	15	X86/Solaris

$$dt_j = st_j + \eta\Box\left[\frac{l_j}{1.1\Box\max(pe)},\frac{l_j}{0.9\min(pe)}\right] \tag{16}$$

Wherein, st_j is the submission time for the task te_i; η is normal distribution random number between the interval (0, 1); l_j is the amount of computation for the task te_i; pe is the processing power of each virtual machine.

4.1.1. Availability

Availability e_r reflects response the first submitting of task te_i, and the probability of successful return before final completion time te_j. On the premise of task te_i selected as t_i, availability e_r satisfies formula (17).

$$e_r=\frac{1}{m}\sum_{j=0}^{m}\sum_{i=0}^{n}\xi_{ij},\xi_{ij}=\begin{cases}1, if \quad te_i \quad success \text{ in } t_i, \\ \qquad 0, otherwise\end{cases} \tag{17}$$

Shown in Figure 1, the availability of CICP algorithm is improve 8.3% under the same conditions with r standard Max-min algorithm and Min-min algorithm, and when the number of users is greater than 400, availability of CICP Algorithm remains between [79%, 83.6 %], which at high level and keeping the stable range.

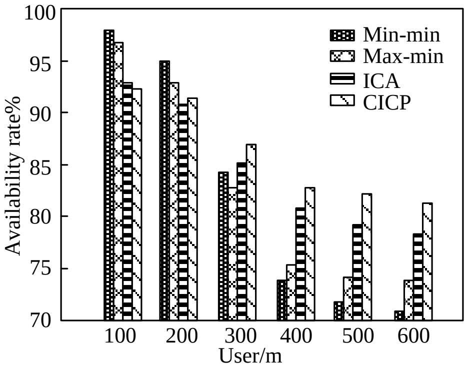

Figure 1. Availability

4.1.2. Load balancing deviation

Load balancing deviation σ reflects the public utilization of resources according to their own abilities and application preferences. To avoid uneven resource loading, load balancing deviation σ satisfies the equation (18)

$$\sigma=\sqrt{\frac{1}{n-1}\Box\sum_{i=0}^{n}(ld_i-ld)^2},i\in\{1,2,...,n\} \tag{18}$$

Wherein, ft_i satisfies the formula (19).

$$ft_i=\frac{c_i}{\sum_{i=0}^{n}c_i}/\frac{l_j}{\sum_{j=0}^{m}l_j},i\in\{1,2,...,n\};j\in\{1,2,...,m\} \tag{19}$$

Wherein, c_i is the cost of resource t_i; l_j is the computation amount of task te_i.

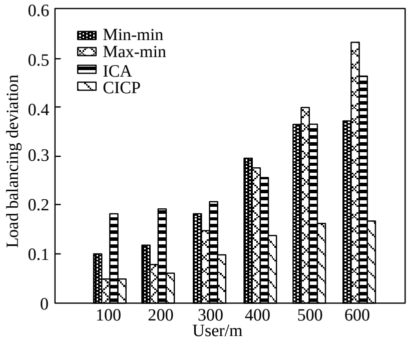

Figure 2. Load balancing deviation

Shown in Figure 2, as the number of users increases, the load balancing deviation is constantly increasing. But compared with Max-min algorithm and Min-min algorithm, load balancing deviation of CICP algorithm, maintains between [0.05, 0.18], which fall 0 28 and achieve a better equitable use of resources. When the number of users is greater than 400, CICP algorithm can maintain at a relatively stable load balancing level.

4.1.3. Effective time

Valid time $\overline{Vt}$ reflects the effective utilization of time when executing tasks in cloud system, which can be shown in the formula (20). Invalid use of time is in the time when the tasks are waiting to be executed..

$$\overline{Vt} = \frac{1}{m}□\sum_{j=0}^{m} \frac{l_j / pe_i}{rdt_j - rst_j}, j \in \{1,2,...,m\}; i \in \{1,2,...,n\} \quad (20)$$

Wherein, rdt and rst_j are respectively the actual submission time and completion time of task te_i; pe_i is the actual processing power of and the assigned resources e_i of task te_i.

Shown in Figure 3, with the number of users are increasing, the effective time $\overline{Vt}$ of users continue to fall. Compared with the Max-min algorithm and Min-min algorithms, the effective time of CICP algorithm is a between [0.68,0.95], which is increased nearly 0 39, so it has a better time utilization.

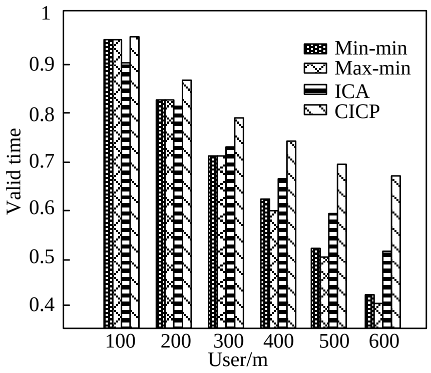

Figure 3. Effective time

5. Conclusion

As a new computing paradigm cloud computing provide QOS services. Resource scheduling is a core and an important challenging component of cloud system. To solve this problem, this paper theoretically makes modeling and analysis on cloud resource scheduling model and proposes CICP resource scheduling algorithm based on performance multi-dimensions QOS of clone immune. Experimental results show that the proposed algorithm meets the scalability and adaptability requirements of cloud computing system, which ensures the effective performance QOS and economic QOS and satisfies the requirements of resource scheduling in cloud computing environments.

References

[1] Muhammad J. Mirza, Nadeem Anjum. Association of Moving Objects Across Visual Sensor Networks. Journal of Multimedia, Vol 7, No 1 (2012) pp. 2-8

[2] Qaisar Rasool, Jianzhong Li, Shuo Zhang, et al.. On P2P and Hybrid Approaches for Replica Placement in Grid Environment. Information Technology Journal, vol. 7. No.4.2008.Pages: 590-598.

[3] Xu CC, Huang X.M., Yang G.W?,et al.. A Two-Layered Replica Management Method. In: the International Joint Conference of IEEE Trust, Security and Privacy in Computing and Communications (TrustCom) and IEEE 10 International Conference on Digital Object Identifier. 2011. Pages: 1431-1436.

[4] eonW., Guptal., and NahrstedtK.. QoS-aware object replication in overlay networks. In: Proceedings of the IEEE GLOBECOM, 2006.

[5] ChenC.,WuJ., and LiuR. QoS-aware, access-efficient, and storage-efficient replica placement in grid environment. Journal of Supercomputing, vol. 49, no. 1, 2009. Pages: 42-63.

[6] GuQ.R,ChenB., ZhangY.R. Dynamic Replica Placement and Location Strategies for Data Grid. In; Proceedings of International Conference on Computer Science and Software Engineering. 2008.

[7] Mohammad Shorfuzzaman, Peter Graham, Rasit Eskicioglu, Distributed placement of replicas in hierarchical data grids with user and system QoS constraints. In; Proceedings of the International Conference on P2P, Parallel, Grid, Cloud and Internet Computing. 2011.

[8] Songlin Bai. The Performance Study on Several Distributed File Systems. In proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011. Pages: 226-229.

[9] Tom White, Hadoop: The Definitive Guide. O'REILLY, 2009.

[10] Vilobh Meshram, Xavier Besseron,Xiangyong Ouyang, et al. Can a Decentralized Metadata Service Layer benefit Parallel Filesystems? In: Proceedings of the IEEE International Conference on Cluster Computing. 2011.