A Universal Fairness Evaluation Framework for Resource Allocation in Cloud Computing

LU Di, MA Jianfeng, XI Ning

Shanxi Key Laboratory of Network and System Security, Xidian University, Xi'an 710071, China

Abstract: In cloud computing, fairness is one of the most significant indicators to evaluate resource allocation algorithms, which reveals whether each user is allocated as much as that of all other users having the same bottleneck. However, how fair an allocation algorithm is remains an urgent issue. In this paper, we propose Dynamic Evaluation Framework for Fairness (DEFF), a framework to evaluate the fairness of an resource allocation algorithm. In our framework, two sub-models, *Dynamic Demand Model* (DDM) and *Dynamic Node Model* (DNM), are proposed to describe the dynamic characteristics of resource demand and the computing node number under cloud computing environment. Combining Fairness on Dominant Shares and the two sub-models above, we finally obtain DEFF. In our experiment, we adopt several typical resource allocation algorithms to prove the effectiveness on fairness evaluation by using the DEFF framework.

Keywords: resource allocation; fairness evaluation; cloud computing

I. INTRODUCTION

In cloud computing, computational resources are highly integrated in the "cloud". Services and applications are provided by virtual machines running over the cloud platform. Hence, computational resources, such as CPU, RAM, bandwidth etc., should be properly scheduled for better service provision.

Resource allocation algorithm is widely studied in recent works on shared communication and computing systems. *max-min fairness*[4][6][15] ensures the allocations of the users with minimal resource demands. In *proportional fairness*[10][14], it attempts to find a balance point in resource allocation among the competing interests. α-*fairness*[19] attempts to determine an equilibrium point between allocation fairness and the utilization efficiency of resources. Ref.[17] presents a game theory based approach which introduces a tradeoff between relay fairness and system throughput. In multi-type resource allocation, ref.[1][3][18] and ref.[5][11][13] focus on multiple instances of the same resource. Ref.[7] proposes *Dominant Resource Fairness* (DRF) which is designed to ensure the fairness in the allocation of multiple types of resources, such as CPU, RAM and bandwidth etc. [2][8] propose genetic algorithm based approaches to obtain the optimal allocation.

Although abundant allocation algorithms are proposed for resource allocation in cloud computing, how to evaluate the fairness of an allocation approach is less studied. Ref.[12] proposes an fairness evaluation model for single-type resource allocation algorithm. Based on [7] and [12], [9] presents a DRF based unified framework, named *Fairness on Domi-*

nant Shares (FDS), for fairness evaluation, in which the efficiency of resource utilization is also considered. In FDS, two key factors are introduced, β and λ. β indicates the fairness type and λ emphasizes the resource utilization (efficiency).

However, in cloud computing, the resource demands of the computing nodes (virtual machines) can vary at different task phases. We define a task phase as a period in which a node is executing one computing task. For example, when the platform is solving equations in different sizes concurrently with computing nodes, the node number can be different according to the size and complexity of the equations. Moreover, the nodes which finish tasks will be terminated, and occupied resources can be released, whereas new nodes will be created for new tasks, and new resource allocations begins. Hence, the resource demand and the node number can change in different period under cloud environment. Both of these dynamic features in cloud are not adequately considered in existing research works.

To address the two issues, we propose a Dynamic Evaluation Framework for Fairness (DEFF) in resource allocation. Our model contains two sub-models, Dynamic Demand Model (DDM) and Dynamic Node Model (DNM). The previous depicts resource demand of the nodes in each task phase, whereas the later gives a description to the variation of node number. With combination of DDM and DNM, we obtain our evaluation model DEFF, which can better adapt the cloud environment.

The rest of the paper is organized as follows. In section II, DDM and DNM are proposed, then DEFF is deduced. In section III the simulations on the DDM, DNM and DEFF are proposed and analyzed. Finally, we state concluding remarks in section IV.

II. System Model

In this section, we firstly give a brief introduction to the FDS evaluation framework. Then, DDM and DNM are proposed for dynamic resource demand and computing node in cloud environment. Finally, with the combination of DDM and DNM, we establish our fairness evaluation framework, DEFF.

2.1 FDS framework

Ref.[7] gives the definition of the maximum share of a resource required by user j to process one computational task as following:

$$\mu_j = \max_i \left\{ \frac{d_{ij}}{R_i} \right\}.$$

d_{ij} denotes the demand on resource i, and R_i indicates the resource capacity. Assume the number of jobs allocated to j is x_j, thus, the Dominant Share (DS) of node j can be denoted as $\mu_j x_j$, and the corresponding resource is dominant resource. Here, a job can be considered as one of a node's execution threads to finish the computing task. Based on DS and max-min fairness, [7] presents dominant share fairness (DRF) allocation algorithm, which determines the node's share of each resource according to its dominant share. Assume the dominant share of node k is maximized, then, its job number reaches to the maximum. Thus, other resource demands of k, which are less than dominant resource, are also satisfied and maximized. According to DRF, all nodes' demands can be maximized without damaging others' interests, hence, allocation based on DS can obtain higher fairness than max-min fairness[7].

Upon [7][12], [9] proposes FDS, which is a dominant share based fairness evaluation framework. FDS is defined as following

$$\mathcal{F}_{\beta,\lambda} = sgn(1-\beta)\left(\sum_{j=1}^{n}\left(\frac{\mu_j x_j}{\sum_{k=1}^{n}\mu_k x_k}\right)^{1-\beta}\right)^{\frac{1}{\beta}}\left(\sum_{j=1}^{n}\mu_j x_j\right)^{\lambda} \quad (1)$$

Equation (1) gives a function, $\mathcal{F}_{\beta,\lambda}$, for evaluating fairness of allocation algorithm as per node's dominant share. $\mathcal{F}_{\beta,\lambda}$ can be divided into two parts: 1) fairness and 2) efficiency. Compared with the fairness function developed in [12], the resource utilization (efficiency) is taken into consideration in $\mathcal{F}_{\beta,\lambda}$, which indicates the adequacy of resource utilization.

This paper proposes Dynamic Evaluation Framework for Fairness (DEFF), a framework for dynamic evaluation of fairness based on dominant share.

This is of importance in the assessment of a fair allocation algorithm. In $\mathcal{F}_{\beta,\lambda}$, $\beta \in \mathbb{R}$ is used to designate the fairness type, whereas $\lambda \in \mathbb{R}$ emphasizes efficiency of resource utilization. By adjusting β and λ, we can obtain various evaluation functions with different fairness type. E.g. as $\beta \to \infty$ and $\lambda = \frac{1-\beta}{\beta}$, $\mathcal{F}_{\beta,\lambda}$ then approaches "*max-min fairness*" on the dominant shares. Moreover, if we take $\beta > 0$ and $\lambda = \frac{1-\beta}{\beta}$, we recover *"α-fairness"*. In particular, taking the limit as $\beta \to 1$ yields *"proportional fairness"*[9]. Without loss of generality, assume the node has only one job (x_j=1), equation (1) can be reduced to

$$\mathcal{F}_{\beta,\lambda} = sgn(1-\beta)\left(\sum_{j=1}^{n}\left(\frac{\mu_j}{\sum_{k=1}^{n}\mu_k}\right)^{1-\beta}\right)^{\frac{1}{\beta}}\left(\sum_{j=1}^{n}\mu_j\right)^{\lambda} \quad (2)$$

Thus, equation (2) provides a universal framework for fairness evaluation. As mentioned before, the nodes' resource demands and their number can vary according to different tasks phases that are not considered in the evaluation framework (equation 2). To address this issue, in the following sections, we firstly propose DDM and DNM model, then we establish our evaluation framework DEFF.

2.2 Dynamic demand model

Assume platform $\mathcal{P}$ with m kinds of resource, node j requires d_{ij} on resource i, thus, we obtain dominant share of j as the definition of μ_j. Obviously, μ_j is determined by the maximal demand of j. However, in different task phases, the dominant share of the nodes can be changed. Hence, μ_j should be the function of time (task phase), which can be redefined as

$$\mu_j(t) = \max_i \left\{\frac{d_{ij}(t)}{R_i}\right\} \quad (3)$$

Assume resource demand of j can be denoted as a vector $\mathbf{D}_j=(d_{1j}, d_{2j}, \ldots, d_{mj})$, and the actual allocation (a_{ij}) may not fulfill the node's requirements, thus, we have $a_{ij} \le d_{ij}$. By introducing a coefficient v_j,($0<v_j \le 1$), we have $\mathbf{a}_j=v_j\mathbf{D}_j$. Since these parameters can change with time, the actual allocation vector can be denoted as

$$\mathbf{a}_j(t)=\mathbf{D}_j(t)\cdot v_j(t) \quad (4)$$

With equation (4), the amount of actual allocation on resource i can be denoted as $a_{ij}(t)=d_{ij}(t)\cdot v_j(t)$. If the capacity of i is R_i, the total allocation of i at time point t_p should satisfy the following relation

$$\sum_{j}^{n} a_{ij}(t_p) = \sum_{j}^{n} d_{ij}(t_p)v_j(t_p) \leqslant R_i \quad (5)$$

Thus, equation (4) gives time based allocation model under dynamic resource demand, which is subjected to equation (5).

2.3 Dynamic node model

When some computing tasks are finished, the relevant nodes will be terminated and the used resources will be released, whereas new node can be created with resource allocation as new tasks are designated. In this case, the total system resource available varies according to the termination and creation of the nodes.

Consider arbitrary time t_p and resource i, we define the set of terminated and created nodes as $\Phi(t_p) = \{node_j | \hat{v}_j(t_p) = 0\}$ and $\Omega(t_p) = \{node'_k | v'_k(t_p) \neq 0 \wedge v'_k(t_p) \notin \mathbf{v}_j(t_p)\}$ respectively. Consider the node number can change with time, let $n(t_p)$ denotes the total number of nodes at any time point t_p. Thus we have

$$\sum_{k} d'_{ik}(t_p)v'_k(t_p) + \sum_{j}^{n(t_p)} d_{ij}(t_p)v_j(t_p) \leqslant R_i \quad (6)$$

Taking into account the creation and termination of the nodes, the amount available of resource i can be denoted as $U_i(t_p) = U_i(t_{p'}) - C_i^{\Omega}(t_p) + C_i^{\Phi}(t_p)$, in which t'_p denotes the previous time point, $C_i^{\Omega}(t_p)$ denotes the total allocation on i for new created nodes, whereas $C_i^{\Phi}(t_p)$ denotes the amount of released resource i. We define the amount of occupied resources at time t_p as $O_i(t_p)$, then $O_i(t_p) = O_i(t'_p) + C_i^{\Omega}(t_p) - C_i^{\Phi}(t_p)$, and $R_i = U_i(t_p) + O_i(t_p)$. Since R_i is a constant, with $U_i(t)$ increases, $O_i(t)$ inevitably decreases, and vice versa. Particularly, as $U_i(t)$=0, resource i is saturated at t_p.

Assume the probability of new node creation at a certain time point is p_ω, whereas that of the node termination is p_φ, define function $g_1(t)$ and $g_2(t)$ as

$$g_1(t)=\begin{cases}1, & in\ p_\omega\\ 0, & in\ 1-p_\omega\end{cases},\quad g_2(t)=\begin{cases}1, & in\ p_\phi\\ 0, & in\ 1-p_\phi\end{cases}. \tag{7}$$

The occupancy of resource i at t_p can be rewritten as

$$O_i(t_p)=O_i(t'_p)+g_1(t_p)\cdot\sum_k^{\|\Omega(t)\|} d'_{ik}(t_p)v'_k(t_p) \\ -g_2(t_p)\cdot\sum_k^{\|\Phi(t)\|}\hat{d}_{ik}(t_p)\hat{v}_k(t_p) \tag{8}$$

In equation (8), $O_i(t'_p)$ is the resource available at previous time point, we have $O_i(t'_p)=\sum_j^{n(t_-)} d_{ij}(t'_p)v_j(t'_p)$ according to equation (4). Thus, equation (8) can be transformed into

$$O_i(t_p)=\sum_j^{n(t_-)} d_{ij}(t'_p)v_j(t'_p)+g_1(t_p)\cdot\sum_k^{\|\Omega(t)\|} v'_k(t_p)d'_{ik}(t_p) \\ -g_2(t_p)\cdot\sum_k^{\|\Phi(t)\|}\hat{v}_k(t_p)\hat{d}_{ik}(t_p) \tag{9}$$

Equation (9) gives dynamic model of "time—probability" for occupancy of resource i, which can reflect the actual features of occupied system resources in the case of dynamic node quantity. Similarly, since the dominant share is the maximal element in the demand vector, the total amount of system dominant shares has the similar form defined as

$$M(t_p)=\sum_j^{n(t_-)}\mu_j(t'_p)v_j(t'_p)+g_1(t_p)\sum_k^{\|\Omega(t)\|}\mu'_k(t_p)v'_k(t_p) \\ -g_2(t_p))\sum_k^{\|\Phi(t)\|}\hat{\mu}'_k(t_p)\hat{v}'_k(t_p) \tag{10}$$

Referring to the definition of $\mu_j(t_p)$, we have $M(t_p)<\sum_i^m O_i(t_p)$. Consider equation (2) and (10), we can further deduce our evaluation model DEFF.

2.4 DEFF model

By the definition of $\mu_j(t)$, the actual amount of dominant share of node j is $\mu_j(t)\cdot v_j(t)$. Thus, equation (2) can be transformed into

$$\mathcal{F}_{\beta,\lambda}(t)=sgn(1-\beta)\left(\sum_{j=1}^{n(t)}\left(\frac{\mu_j(t)v_j(t)}{\sum_{k=1}^{n(t)}\mu_k(t)v_k(t)}\right)^{1-\beta}\right)^{\frac{1}{\beta}}\left(\sum_{j=1}^{n(t)}\mu_j(t)v_j(t)\right)^{\lambda} \tag{11}$$

Further, if $\sum_{k=1}^{n(t)}\mu_k(t)v_k(t)$ denotes the sum of system dominant share, with the consideration of node number variation and equation (10), we can obtain the following definition.

$$\mathcal{F}_{\beta,\lambda}(t)=sgn(1-\beta)\left(\sum_{j=1}^{\eta(t)}\left(\frac{\mu_j(t)v_j(t)}{M(t)}\right)^{1-\beta}\right)^{\frac{1}{\beta}}(M(t))^{\lambda} \tag{12}$$

Equation (12) is our fairness evaluation framework (DEFF), in which $\eta(t)$ denotes the node number at t, and we have $\eta(t)=n(t)+\|\Omega(t)\|-\|\Phi(t)\|$. It can be obtained from DEFF model that the node number and the total dominant share can vary with time. Hence, equation (12) can reflect the influences to allocation fairness ($\mathcal{F}_{\beta,\lambda}(t)$), which are brought by such variations.

Theorem 1: DEFF does not change the properties of prototype evaluation model.

Proof. For any time point t_p, the system node number can be considered as an instantaneous constant n_{t_-}. Thus, the total dominant share become a constant denoted as $M(t_p)=\sum_j^{n_-}\mu_j(t_p)v_j(t_p)$. Since $\mu_j(t_p)$ and $v_j(t_p)$ are determined at this moment, we have $M(t_p)=\sum_j^{n_-}\mu_j v_j$, hence,

$$\mathcal{F}_{\beta,\lambda}(t_p)=sgn(1-\beta)\left(\sum_{j=1}^{n_-}\left(\frac{\mu_j v_j}{\sum_j^{n_-}\mu_j v_j}\right)^{1-\beta}\right)^{\frac{1}{\beta}}\left(\sum_{j=1}^{n_-}\mu_j v_j\right)^{\lambda},$$

that is, equation (12) is regressed to the form of equation (2) as v_j=1. In summary, equation (12) can be transformed into the prototype evaluation framework at a determined moment. On the other hand, equation (2) can be considered as the special case of the model (12) at a determined moment.

III. ANALYSIS AND EVALUATION

In this section, we discuss the DDM and DNM model, and give the simulations and analysis on DEFF.

3.1 Analysis of DDM

For arbitrary resource i, according to equation (4), the total allocation can be denoted as

$$s_i(t) = \sum_j^{n(t)} d_{ij}(t) v_j(t),\ (s_i \leqslant R_i),$$

which can be defined as

$$\mathcal{L}_t:\ \sum_j^{n(t)} d_{ij}(t) v_j(t) - s_i(t) = 0 \tag{13}$$

$\mathcal{L}_t$ determines a hyperplane whose normal vector $\mathbf{d}_i(t)$=($d_{i1}(t)$,$d_{i2}(t)$, ..., $d_{in(t)}(t)$) is perpendicular to the allocation vector $\mathbf{v}(t)$ and $\mathcal{L}_t$. The intersection of $\mathbf{d}_i(t)$ and $\mathcal{L}_t$ is the solution to resource allocation at t, which is marked as $\mathbf{v}^*$. Furthermore, an allocation algorithm with fairness (e.g. *max-min fairness*) is actually to build a hyperplane $\mathcal{L}_t$ to satisfy certain fairness requirements, and finally find out the optimized $\mathbf{v}^*$ as per the principles of algorithm. For different time t, $t \in [0, \infty)$, there exists a set of demand vector on resource i, denoted by $\mathbf{D}_i$=($\mathbf{d}_i(t_1)$,$\mathbf{d}_i(t_2)$, ..., $\mathbf{d}_i(t_n)$). Each element of $\mathbf{D}_i$ is a demand vector for resource i and satisfy the constraint $\sum \mathbf{d}_i(t_p) \leqslant R_i$. The essence of the resource allocation is to find a hyperplane perpendicular to the normal vector $\mathbf{d}_i(t_p)$ at time t_p according to the fairness principles provided by allocation algorithm, and finally find out the intersection $\mathbf{v}^*(t_p)$.

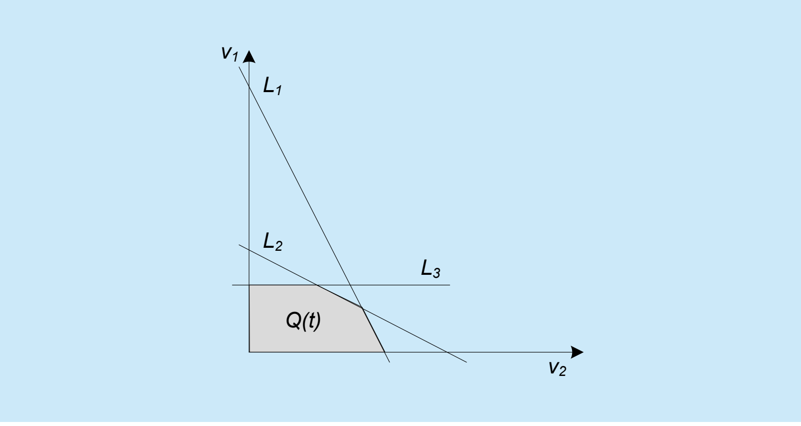

Fig.1 *Solution space under d_1 and d_2*

To illustrate this point, we take an example of allocation issue with 2 nodes and 3 types of resources. Assume the demand vectors of the two node are $\mathbf{d}_1 = (\frac{1}{4}, \frac{2}{3}, 1)$ and $\mathbf{d}_2 = (\frac{1}{2}, \frac{1}{3}, 0)$, if the allocation result is v=(v_1,v_2), then the following equations hold.

$$\begin{cases} \frac{1}{4}v_1 + \frac{1}{2}v_2 \leqslant 1:\ L_1 \\ \frac{2}{3}v_1 + \frac{1}{3}v_2 \leqslant 1:\ L_2 \\ v_1 \leqslant 1:\ L_3 \end{cases} \tag{14}$$

(14) defines three linear equations, L_1, L_2 and L_3, which form a closed area shown as figure 1. The gray area is the solution space $Q(t)$ formed by three lines according to the resource constraints. With certain fair allocation algorithm, we obtain a point (v_1^*, v_2^*) in $Q(t)$ area, which is the solution to the fair allocation problem. Different allocation algorithms can generate different solutions, however, the set of solutions is definitely in $Q(t)$. Since resource demand vector changes over time, L_1, L_2, L_3 and the rectilinear region can change at different time, nevertheless, the allocation solution is still in $Q(t)$. Thus, definite algorithm and demand vector (at a definite time) can uniquely determine $Q(t)$ and the solution.

Consider a scenario with n nodes and m resources, the lines in the example above become n-dimension hyperplanes, and the demand vectors become the normal vectors perpendicular to these hyperplanes. The solution space, marked as $\mathcal{S}(t)$, is constructed by m n-dimension hyperplanes, in which the solution is located. Note that $\mathcal{S}(t)$ is also the function of time.

3.2 Analysis of DNM

From equation (9), the occupied resource i at a certain time, $O_i(t)$, consists of three parts: 1) the previous occupancy, $O_i(t')$, 2) the new allocation at this time, $C_i^{\Omega}(t)$, and 3) the amount of released resource $C_i^{\Phi}(t)$. Since the released resource must be from the occupancy at the previous time, we have the constraint $O_i(t') \geqslant C_i^{\Phi}(t)$. Once the equality holds, it implies that the system releases all resources allocated in the previous time t', thus, we have $O_i(t) \geq 0$. Similarly, the amount of spare resource i at t, $U_i(t)$, is also composed of three parts: 1) the free resources in the previous time, $U_i(t')$, 2) the new allocation $C_i^{\Omega}(t)$, and 3) the released amount $C_i^{\Phi}(t)$ at the current time. If $U_i(t)=0$, it implies that the system has no spare resource at t, and $O_i(t)$ reaches its maximum. Otherwise, if $O_i(t)=0$, $U_i(t)$ reaches the maximal value.

To give a further discussion on the relation between $O_i(t)$ and $U_i(t)$, the capacity of resource i, R_i, is normalized as $R_i=1$, then, according to section 2.3, we have $O_i(t)+U_i(t)=1$. For example, in figure 2, assume $O_i(t')=0.3$ at t', then $U_i(t')=0.7$ (shown as point $p_{t'}$). Thus, we can obtain from figure 2 that $O_i(t)$ and $U_i(t)$ are determined by L. In other word, an arbitrary point on L are a pair of possible $O_i(t)$ and $U_i(t)$. We define a family of linear functions f as straight lines passing through the origin and intersect with L, then, the set of intersections is the set of possible values of $O_i(t)$ and $U_i(t)$. Let $k = f' = \frac{O_i(t)}{U_i(t)}$, we have the following relations,

1) $k \to 0$: $O_i(t) \to 0$, the system resource available reaches to the maximum, $U_i^{max} = R_i$;

2) $k \to \infty$: $U_i(t) \to 0$, the occupied system resource reaches to the maximum, $U_i^{max} = R_i$;

3) $k=1$: $U_i(t)=O_i(t)$, the resource occupancy equals to the spare (e.g. point p_m in figure 2).

$k=1$ is a special case, that in this moment, f and L intersect at point p_m. If $O_i(t)$ and $U_i(t)$ are at the lower half of L below p_m ($0 \leq k < 1$), the spare resource i is more than the occupied. That is, at that moment, the amount of resource i can still satisfy the future demand. If $O_i(t)$ and $U_i(t)$ are at the upper half of L above p_m ($1 \leq k < \infty$), in this case, the occupancy of resource i is more than the spare. Moreover, with the increase of k, i can be the bottleneck, and the system has to adjust the allocation algorithm to prevent i from being saturated earlier.

As mentioned before, $O_i(t)$ and $U_i(t)$ are actually determined by $C_i^{\Omega}(t)$ and $C_i^{\Phi}(t)$, hence, k is indirectly determined by these two values. Thus we have the following relations,

1) $k\uparrow$: caused by more allocation and less release, that implies $C_i^{\Omega}(t) > C_i^{\Phi}(t)$ in most time;

2) $k\downarrow$: caused by more release and less allocation, that implies $C_i^{\Omega}(t) < C_i^{\Phi}(t)$ in most time.

In summary, to prevent any resource from being the bottleneck too quickly, the scheduler should control $C_i^{\Omega}(t)$ and $C_i^{\Phi}(t)$ according to k, that ensures the availability of the system resources.

3.3 Evaluations of DEFF Model

In this section, we evaluate the effectiveness of DEFF model. Firstly, to prove the effectiveness of using DEFF on fairness evaluation, we adopt DRF and Max-min algorithms as examples in our experiments. Then, we adopt a utility based algorithm, α-*fairness*, to show the effectiveness of evaluating fairness variation by using DEFF when adjusting α factor (since proportional fairness is also a utility based algorithm similar with α-*fairness*, we take α-*fairness* as an example).

In our first experiment, we use DEFF to show the difference on fairness between DRF and Max-min allocation algorithms. Let $\beta \in (0,20)$, $\lambda = \frac{1-\beta}{\beta}$, $p_\omega=0.6$, $p_\phi=0.7$, the re-

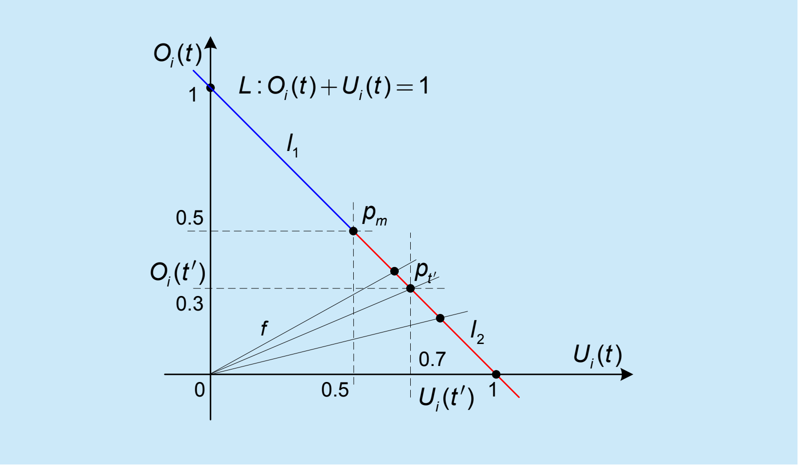

Fig.2 *The relation between $O_i(t)$ and $U_i(t)$*

source capacity R={300,400,250}, initial node number n=20, 0<d_{ij}≤30, t∈[0,50], demand vector of i, $\mathbf{D}_i(t)$,(i=1,2, ..., n), can make a n-dimension solution space similar to Figure 1.

Figure 3 shows t-based DEFF model, in which "*running times*" indicates t variation, the surface in blue denotes the fairness value with DRF, whereas the dark gray is that of using max-min. Obviously, with DEFF model, $\forall p \in S1$ and $\forall q \in S2$, we have $p>q$. Thus, according to [7][9], DRF can achieve better fairness values than that of using max-min. This further implies that under the variations of time and resource demands, DEFF model can still preferably provide the measurement on allocation fairness.

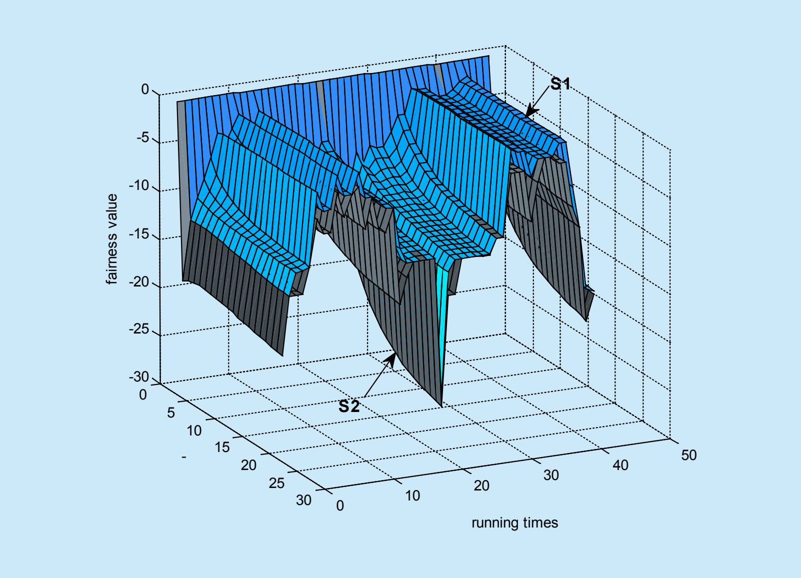

Fig.3 *Evaluation of DEFF Model (using DRF & Max-min)*

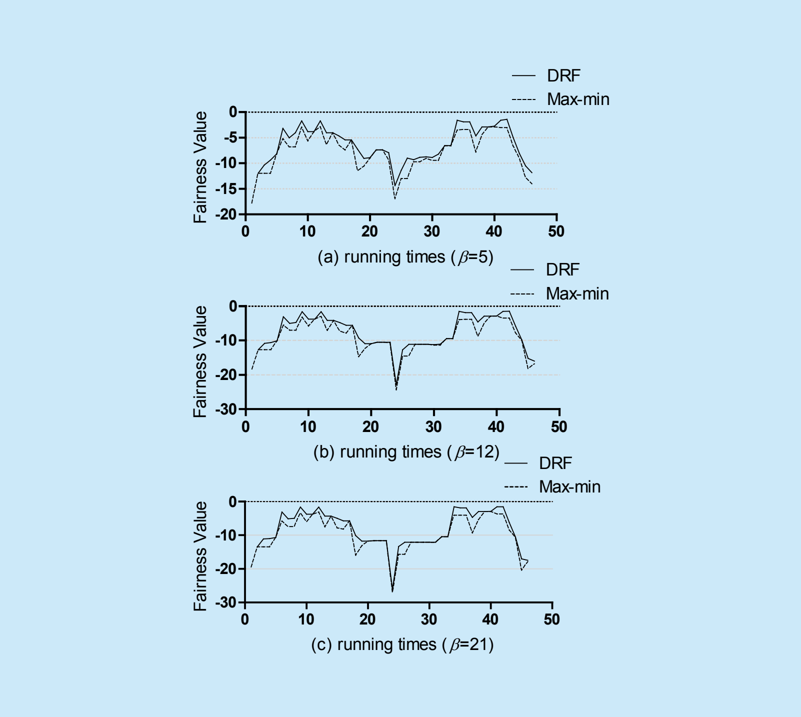

Fig.4 *Fairness variation as β=5,12,21*

Figure 4 shows the sample slices of figure 3 along the axis "*running times*", in which abscissa is the time variation, whereas the ordinate is the fairness value, and the sample points β=5,12,21. It can be obtained that the curve of using DRF always larger than that of using max-min. Hence, with DEFF, we can see that DRF has higher fairness than max-min as the factorβis determined.

Figure 5 shows three sample slices from Figure 3 on theβ-axis. Since the node number varies according to p_ω and p_ϕ, the total dominant share presents dynamic characteristics. With the changes of fairness factor β, DRF can always achieve better fairness than max-min algorithm. When β is getting larger, the DRF curve has moderate changes, whereas the max-min curve decreases more rapidly. This reveals DRF has higher stability than max-min.

In our second experiment, the effectiveness of DEFF on evaluating the α-fairness is presented. α-fairness is actually an adjusting algorithm, which attempts to find an equilibrium point between efficiency (*utilization* or *revenue*) and fairness[19], hence, in our experiment, we only show the fairness variation when adjusting α (discussing efficiency is beyond the scope of this paper).

Let p_ω=0.6, p_ϕ=0.7, α∈[0,1], the resource capacity R={600}. Initial node number n=10, 5≤d_{ij}≤25, t∈[0,50]. We use Max-min algorithm to allocate resource, and adjust the allocation results with α-fairness.

Figure 6 shows the fairness variation when adjusting α at each running time. Although running time t∈[0,50], because when the node number changes, at some running time points, no node exists (*null* points). Therefore, in this figure, we have only 27 running time points after removing *null* points. We can see that as α changes from 0 to 1, the fairness value decreases at each running time. However, for some running times, fairness values decrease rapidly, whereas some decrease slowly or even have very unobvious changes. We will choose some running times as examples to discuss the details on fairness variation.

Figure **7** shows two typical examples of the fairness variation under different allocation vectors with changes of α.

For Figure **7**(a), *running time t*=27, the allocation vector A_{27}={180,200,220}, and the total occupied resource is $O_{27} = \sum A_{27} = 600$. According to α-*fairness*, $A_{27}^{\alpha=0} = \{200, 200, 200\}$ α=0 and $A_{27}^{\alpha=1} = \{180, 200, 220\}$, α=1. $A_{27}^{\alpha=0}$ is totally fair, as each node gets the same amount of resource, however, each node cannot be satisfied according to its resource demand (some need more, but some need less). $A_{27}^{\alpha=1}$ is less fair than $A_{27}^{\alpha=0}$, however, the nodes' resource demands are fully considered. Hence, with the curve generated by DEFF, we can try to find an equilibrium point between the fairness and nodes' demands. This is beyond the scope of this paper.

Besides, the elements of $A_{27}^{\alpha=1}$ approximate to that of $A_{27}^{\alpha=0}$, hence, changing α only leads slight decline on fairness (about -3.0 to -3.04). Hence, the curve seems nearly flat. For Figure **7**(b), running time t=15, the allocation vector A_{15}={82.67, 25.0, 246.5, 47.67, 98.25}, and the total occupied resource is $O_{15} = \sum A_{15} = 500.09$. When α=0, we obtain the adjusted vector $A_{15}^{\alpha=0} = \{a_i | a_i = 100.018, i = 1, 2, 3, 4, 5\}$, whereas the vector becomes the original one whenα=1 ($A_{15}^{\alpha=1} = A_{15}$). Since $A_{15}^{\alpha=0}$ is much fairer than $A_{15}^{\alpha=1}$ (without considering nodes' resource demands), the curve decreases obviously (about -5 to -11).

In summary, our experiment results reveal that for the typical resource allocation algorithms and utility-based fairness algorithm,

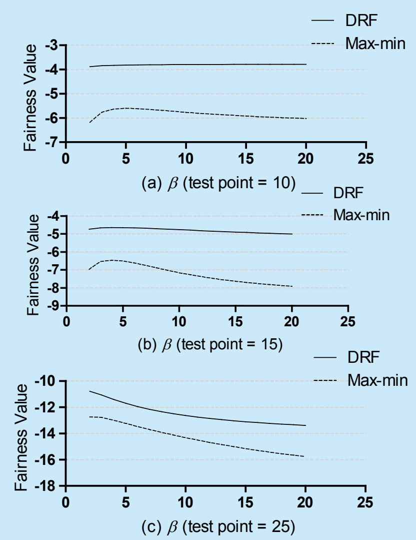

Fig.5 *Fairness variation as t=10,15,25*

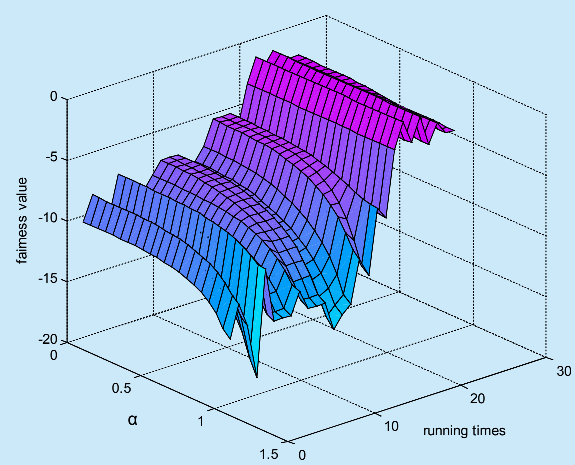

Fig.6 *Evaluation of DEFF Model (using α-fairness)*

DEFF model can effectively evaluate the fairness changes with the time and node number variation.

IV. Conclusions

This paper proposes DEFF, a framework for dynamic evaluation of fairness based on dominant share. Aiming to the dynamic resource demand and computing node in cloud comput-

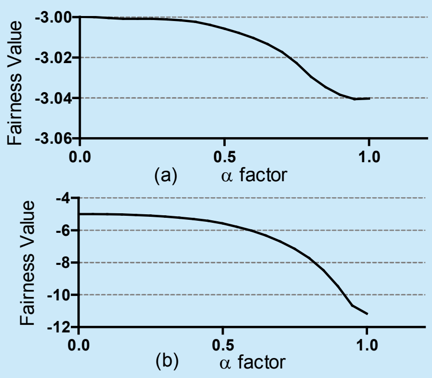

Fig.7 *Fairness variation under different allocation*

ing, we introduce time and probability factor to establish two sub-models: 1) Dynamic Demand Model (DDM) and 2) Dynamic Node Model (DNM). By combining these two model, we establish Dynamic Evaluation Framework for Fairness (DEFF) to give a measurement to resource allocation algorithms. To evaluate the effectiveness of DEFF, we adopt two typical allocation algorithms, DRF and max-min, and a utility-based fairness algorithm, α-fairness in our experiment. According to experiment results, DEFF shows preferably effectiveness under dynamic demand and node number. Our framework provides significant reference for determining resource allocation algorithms in cloud computing.

ACKNOWLEDGEMENTS

The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have helped improve the quality of this paper. This work was supported in part by Program for Changjiang Scholars and Innovative Research Team in University No. IRT1078; The Key Program of NSFC-Guangdong Union Foundation No. U1135002; The Fundamental Research Funds for the Central Universities No. JY0900120301.

References

[1] BARUAH S K, GEHRKE J, PLAXTON C G. Fast scheduling of periodic tasks on multiple resources[C]. IPPS. IEEE Computer Society, 1995, 280-288.

[2] CAMPEGIANI P. A Genetic Algorithm to Solve the Virtual Machines Resources Allocation Problem in Multi-tier Distributed Systems[C]. VPACT'09, 2009.

[3] BARUAH S K, COHEN N K, PLAXTON C G, and Donald A. Varvel. Proportionate Progress: A Notion of Fairness in Resource Allocation[J]. Algorithmica, 1996, 15(6): 600–625.

[4] BERTSEKAS D, GALLAGER R. Data Networks[M]. Prentice Hall, 1992.

[5] BLANQUER J M, ÖZDEN B. Fair Queuing for Aggregated Multiple Links[C]. SIGCOMM, 2001: 189–197.

[6] CHARNY A, CLARK D, JAIN R. Congestion Control with Explicit Rate Indication[C]. International Conference on Communications, 1995(3): 1954-1963.

[7] GHODSI A, ZAHARIA M, HINDMAN B, *et al*. Dominant Resource Fairness: Fair Allocation of Multiple Resource Types[J]. In Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, 2011, 24–24.

[8] GU J H, HU J H, ZHAO T H, *et al*. A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment[J]. Journal of Computers, 2012(7): 42–52.

[9] WONG C J, SEN S, LAN T, *et al*. Multi-resource Allocation: Fairness-Efficiency Tradeoffs in A Unifying Framework[C]. In INFOCOM, IEEE, 2012: 1206–1214.

[10] KELLY F P. Charging and Rate Control for Elastic Traffic[J]. European Transaction on Telecommunications, 1997(8): 33-37.

[11] KLEINBERG J M, RABANI Y, and TARDOS É. Fairness in Routing and Load Balancing[J]. Journal of Computer System Sciences, 2001, 63(1):2–20.

[12] LAN T, KAO D, CHIANG M, *et al*. An Axiomatic Theory of Fairness in Network Resource Allocation[C]. In INFOCOM, IEEE, 2010, 1343–1351.

[13] LIU Y and KNIGHTLY E W. Opportunistic Fair Scheduling over Multiple Wireless Channels[C]. INFOCOM, 2003.

[14] MASSOULIÉ L and ROBERTS J. Bandwidth Sharing: Objectives and Algorithms[C]. INFOCOM, 1999, 1395–1403.

[15] TAN L, PUGH A C and YIN M. Rate-based Congestion Control in ATM Switching Networks Using A Recursive Digital Filter[J]. Control Engineering Practice, 2003, 11(10): 1171–1181.

[16] TIAN J F, YUAN P, and LU Y Z. Security for Resource Allocation Based on Trust and Reputation in Computational Economy Model for Grid[C]. Frontier of Computer Science and Technology 2009, IEEE, 2009: 339–345.

[17] TENG Y L, HUANG T, LIU Y Y, *et al*. Cooperative Game Approach for Scheduling in Two-Virtual-Antenna Cellular Networks with Relay Stations Fairness Consideration[J]. China Communications, 2013, 10(2):56–70.
[18] ZHU D, MOSSÉ D, and MELHEM R G. Multiple-Resource Periodic Scheduling Problem: How Much Fairness is Necessary?[C] RTSS, IEEE Computer Society, 2003: 142–151.
[19] ZUKERMAN M, TAN L S, WANG H W, *et al*. Efficiency-Fairness Tradeoff in Telecommunications Networks[C]. IEEE Communications Letters, 2005: 643–645.

Biographies

LU Di, received the B.S., M.S. and Ph. D. degrees in Computer Science and Technology from Xidian University, China in 2006, 2009 and 2014. Now he is a lecture in school of Computer Science and Technology, Xidian University. His research interests include resource scheduling, virtualization technology, operating system and storage technology. Email: dlu@xidian.edu.cn.

MA Jianfeng, received the MS and Ph.D degree in Xidian University, China in 1989 and 1995 separately. He has been a professor in Department of Computer Science and Technology, Xidian University since 1998. Now he is also the special engaged professor of the Yangtze River scholar in China. He is currently working on the heterogeneous wireless networks convergence, service computing, wireless network security, the survivability of network system and so on. He is the leader of the Key Program of NSFC-Guangdong Union Foundation, National Natural Science Foundation of China and Major national S&T program.

XI Ning, received the B.S, M.S and Ph. D. degrees in Computer Science and Technology from Xidian University, China in 2008, 2011 and 2014. And now he is a lecture in school of Computer Science and Technology, Xidian University. His major research is in home network, service computing and network security.