iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** VI **Month of publication:** June 2023

DOI: https://doi.org/10.22214/ijraset.2023.54373

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VI Jun 2023- Available at www.ijraset.com

A Universal Fairness Evaluation Framework for Resource Allocation in Cloud Computing

Prof. Pragati Patil[1], Chandrashekhar G. Bhagat[2]
[1]Project Guide, Department of Computer Engineering, Abha Gaikwad Patil College of Engineering, Nagpur
[2]PG Scholar, Department of Computer Engineering, Abha Gaikwad Patil College of Engineering, Nagpur

Abstract: Machine learning is nowadays ubiquitous, providing mechanisms for supporting decision making that leverages big data analytics. However, this recent rise in importance of machine learning also raises societal concerns about the dependability and trustworthiness of systems which depend on such automated predictions. 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 etc.

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] 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. A *fairness* 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] 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.

Machine Learning (ML) is nowadays ubiquitous, as most organizations take advantage of it to perform or support decisions within their systems [1], [2]. ML is an area of Artificial Intelligence (AI) in which we use a set of statistical methods and computational algorithms to allow computers to learn from data [3]. ML algorithms can be divided into two main groups: supervised and unsupervised. Supervised learning involves the development of computational models for estimating an output based on previously known inputs and outputs. In unsupervised learning, the models are built based solely on existing inputs but there are no associated outputs that may be used for sake of training.

We may face fairness and transparency issues for both groups of algorithms. It is now commonplace to run ML systems in cloud-based infrastructures, motivated by issues such as elasticity, robustness, and ease of operation [4]. In practice, cloud services are fueling big data analytics, allowing organizations to make better and faster decisions using data that previously were hard or impossible to use [5]. This raises many opportunities in today's competitive environment, by offering many services using highly scalable technologies on a pay-as-you-go basis. However, it also creates new challenges regarding trust, a paramount concern in critical systems [5]. Regulatory institutions have long focused these properties namely in OECD's fair information practices [6] and in EU Privacy Directive 95/46/EC [7]. However, such legislation has never received as much emphasis as now. The new EU General Data Protection Regulation (GDPR) [8] shifts the onus to the organizations, demanding them to demonstrate that they are taking the appropriate measures to protect the legal rights of the individuals and their data, requiring privacy-preserving, fair and transparent systems.

II. PROPOSED SYSTEM

1) From the context of the project we introduce an initial set of techniques that are being developed not just to support but also to monitor and assess fairness and transparency in the context of ML applications and systems. Finally, we present concrete examples of practical application of these techniques in Lemonade, and how they integrate with other components.

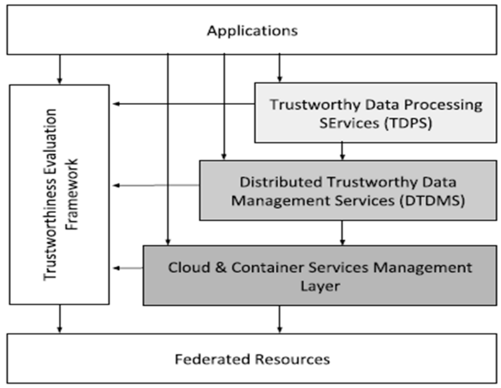

Fig. 1. The model of the ATMOSPHERE project.

2) The application itself or the execution framework for adjusting parameters to increase trust or to react to runtime failures in federated infrastructures, up to the limits on resource allocation that a user may have set - avoiding infinite consumption of resources. Fairness and transparency are mainly monitored at the layer of the data processing service.

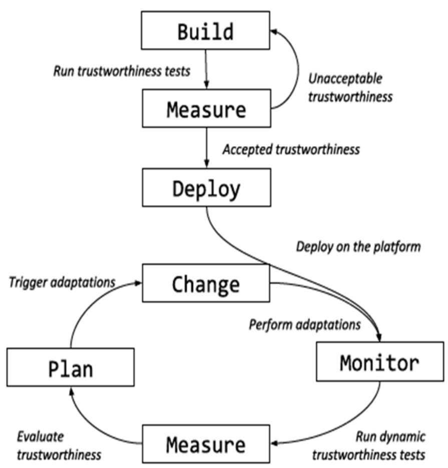

Fig. 2. The lifecycle of ATMOSPHERE applications.

III. 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.

A. 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,

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

dij denotes the demand on resource *i*, and *Ri* indicates the resource capacity. Assume the number of jobs allocated to *j* is *xj*, thus, the Dominant Share (DS) of node *j* can be denoted as μ*jxj* , 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,

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

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 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 $\beta \in \mathbb{R}$ 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 (xj=1), equation (1) can be reduced to,

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

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.

IV. ANALYSIS AND EVALUATION

A. Analysis of DDM

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

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

which can be defined as,

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

L*t* determines a hyperplane whose normal vector **d***i*(*t*)=(*di*1(*t*),*di*2(*t*), ..., *din*(*t*)(*t*)) is perpendicular to the allocation vector **v**(*t*) and L*t* The intersection of **d***i*(*t*) and L*t* is the solution to resource allocation at *t*, which is marked as **v***. Furthermore, an allocation algorithm with fairness (e.g. *max-min fairness*) is actually to build a hyperplane L*t* to satisfy certain fairness requirements, and finally find out the optimized **v*** as per the principles of algorithm.
For different time *t*, *t* _ [0,_) , there exists a set of demand vector on resource *i*, denoted by **D***i*=(**d***i*(*t*1),**d***i*(*t*2), ..., **d***i*(*tn*)). Each element of **D***i* is a demand vector for resource *i* and satisfy the constraint _ d*i*(*tp*) _ *Ri* . The essence of the resource allocation is to find a hyperplane perpendicular to the normal vector **d***i*(*tp*) at time *tp* according to the fairness principles provided by allocation algorithm, and finally find out the intersection **v***(*tp*).

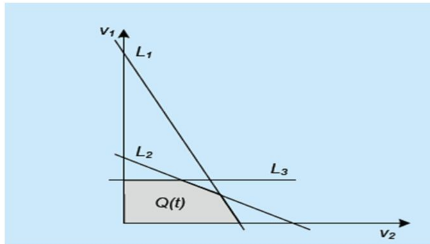

Fig.3 *Solution space under d1 and d2*

B. Analysis of DNM

From equation (9), the occupied resource *i* at a certain time, *Oi*(*t*), consists of three parts:

1) the previous occupancy, *Oi*(*t*'), 2) the new allocation at this time, *C_ i* (*t*) , and 3) the amount of released resource *C_ i* (*t*) . Since the released resource must be from the occupancy at the previous time, we have the constraint *Oi*(*t*_) _ *C_ i* (*t*) . Once the equality holds, it reimplies that the system releases all resources allocated in the previous time *t'*, thus, we have *Oi*(*t*)≥0. Similarly, the amount of spare resource *i* at *t*, *Ui*(*t*), is also composed of three parts: 1) the free resources in the previous time, *Ui*(*t*'),

2) the new allocation *C_ i* (*t*) , and

3) the released amount *C_ i* (*t*) at the current time. If *Ui*(*t*)=0, it implies that the system has no spare resource at *t*, and *Oi*(*t*) reaches its maximum. Otherwise, if *Oi*(*t*)=0, *Ui*(*t*) reaches the maximal value.

C. 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).

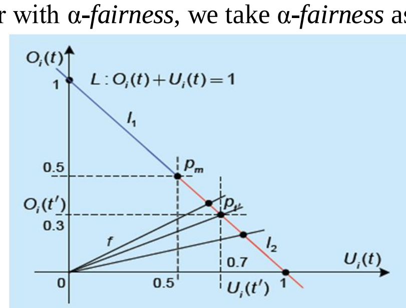

Fig.4 The relation between Oi(t) and Ui(t)

In our first experiment, we use DEFF to show the difference on fairness between DRF and Max-min allocation algorithms. Let $\beta \in (0,20)$, $\lambda = \dfrac{1-\beta}{\beta}$, p_ω=0.6, p_ϕ=0.7, as the resource capacity *R*={300,400,250}, initial node number *n*=20, 0<*dij*≤30, *t*∈[0,50], demand vector of *i*, **D***i*(*t*),(*i*=1,2, ..., *n*), can make a *n*-dimension solution space similar to Figure 3.

Figure 5. shows *t*-based DEFF model, in which "*running times*" indicates *t* variation.

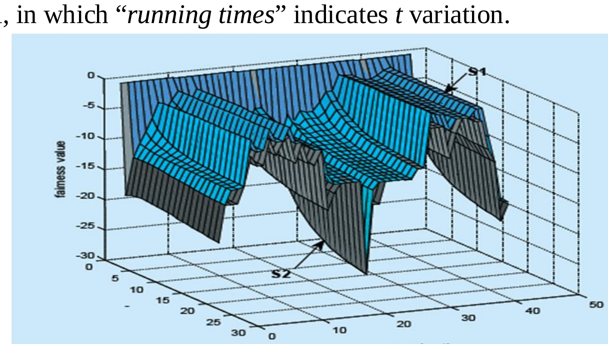

Fig.5 Evaluation of DEFF Model (using DRF & Max-min)

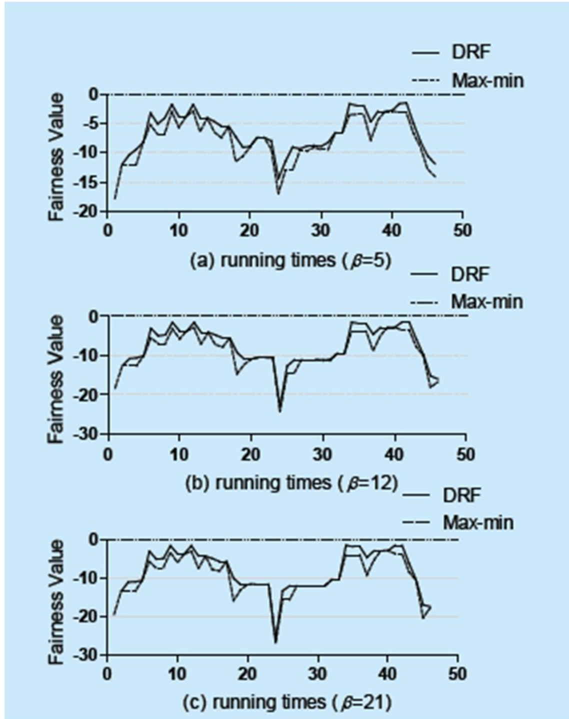

Fig.6 Fairness variation as β=5,12,21

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.

Figure 6 shows the sample slices of figure 5 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 7 shows three sample slices from Figure 5 on theβ-axis. Since the node number varies according to $p\omega$ and $p\phi$, 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 maxmin 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 $p\omega$=0.6, $p\phi$=0.7, $\alpha \in [0,1]$, the resource capacity R={600}. Initial node number n=10, $5 \leq dij \leq 25$, $t \in [0,50]$. We use Max-min algorithm to allocate resource, and adjust the allocation results with α-fairness.

Figure 8. shows the fairness variation when adjusting α at each running time. Although running time $t \in [0,50]$, because when the node number changes, at some running time points, no node exists (*null* points). Therefore, in this computfigure, 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 **9** shows two typical examples of the fairness variation under different allocation vectors with changes of α.

For Figure **9**(a), *running time* t=27, the allocation

vector $A27$={180,200,220}, and the total

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VI Jun 2023- Available at www.ijraset.com

occupied resource is, $O_{27} = \sum A_{27} = 600$
According to α-*fairness*, *A*_=0
= {200, 200, 200}
α=0 and *A*_=1
= {180, 200, 220} , *α*=1. *A*_=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*_=1 is less fair than *A*_=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*_=1 approximate to that of *A*_=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 i s, $O_{15} = \sum A_{15} = 500.09$
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*_=1
15 = *A*15). Since *A*_=0 15 is much fairer than *A*_=1
15 (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,

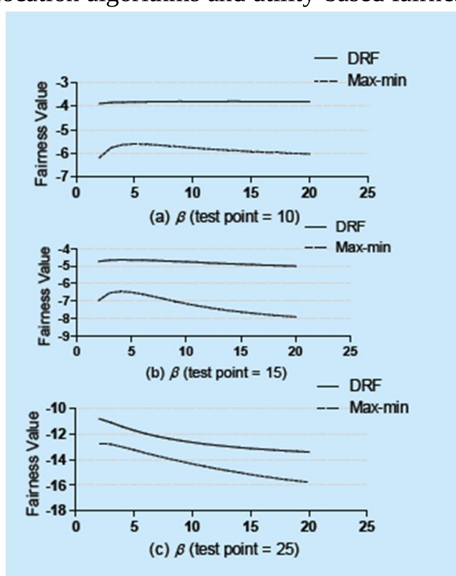

Fig.7 Fairness variation as t=10,15,25

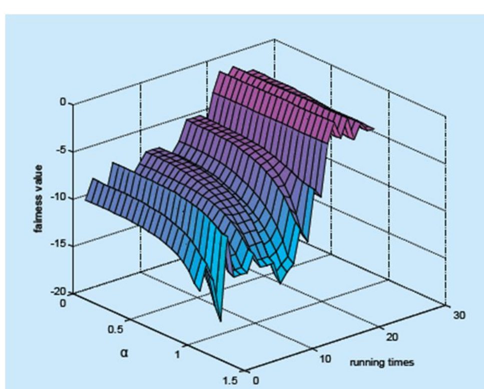

Fig.8 Evaluation of DEFF Model (using α-fairness)

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

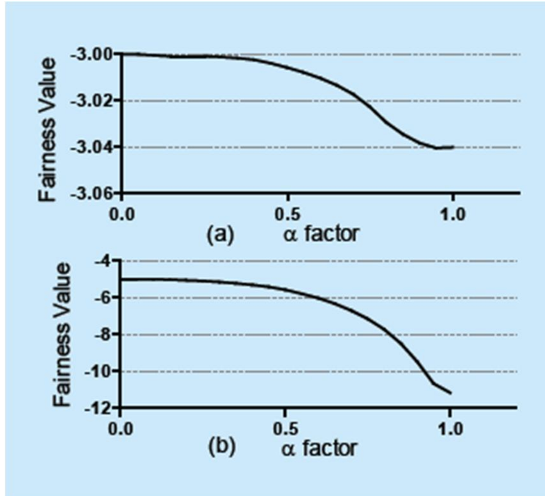

Fig.9 Fairness variation under different allocation

V. CONCLUSION

In this work, a framework for evaluating two cloud pricing strategies–bundled and resource pricing–in terms of their resulting fairness and revenue. We first characterize client demand for resources as a function of the prices offered under these different pricing plans. After showing some analytical bounds on the tradeoff between fairness and revenue, we compare achieved fairness and revenue under the two pricing plans. We finally use data taken from a Google cluster to numerically evaluate the impact of resource capacity and volume discounts on the operator's fairness-revenue tradeoff.

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 computfigure 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.

VI. FUTURE WORK

Future extensions of this work will consider an additional pricing scheme: differentiated pricing, in which the operator can choose a per-job price for each client independent of the client's resource requirements. We do not consider such a pricing plan here since bundled and resource pricing are more practically relevant; in practice clients are generally not charged different per-job prices. One could also extend our work to take into account job completion deadlines, which impose an additional constraint on the resources allocated at any given time. We also plan to consider tradeoffs between revenue, fairness, and operational efficiency, e.g., through examining the total amount of leftover resources.

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.