A Research on Resource Allocation Solution with Cost Optimization in Cloud Computing System

Zhou FANG[1], Buyang CAO[1], and Xu JIANG[2]
[1]School of Software Engineering, Tongji University, Shanghai, China
[2]eBay COE, Shanghai, China

Abstract - *In this paper, a cost saving based allocation strategy for infrastructure resource allocation is proposed to support the business service in cloud computing environment. In a typical enterprise cloud system, it needs the capability to automatically allocate resources by matching resource requirements upon resource availability, while taking both load-balance and cost saving into account. On the foundation of base allocation strategy which is focus on only load-balance consideration, we propose a series of steps by using normalize both workload and cost value of infrastructure resource to enhance the base strategy taking the consideration of cost saving. These steps lead to an improved resource allocation strategy aiming balancing the workload and keeping operational cost low.*

Keyword: cloud computing, resource allocation, load-balance, cost saving, normalization

1 Introduction

The technology of "Cloud" has been an active research topic in recently years. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. The Cloud model has one of essential characteristics: cloud Infrastructure as a Service (IaaS). This model provides the capability (resources) for processing, storage, networks, and other fundamental computing requests without managing the underlying cloud infrastructure by a resource requestor himself. In most cloud computing systems, the infrastructure resources are loosely coupled and distributed. What's more, with the popularity of cloud computing concept, the virtualization is applied as one of essential technologies to build cloud infrastructure. To leverage the virtualization, service providers is able to build up cloud computing platforms not only with less investments but also fully utilizing existing infrastructure capacities. These technologies enable different services to run in a virtually isolated environment and allow resources that are allocated to these services to scale up and down upon demands transparently and seamlessly [1][2][6][10].

These benefits of cloud computing attract more enterprises to migrate their internal/external services or applications to cloud platforms, the departments of cloud computing service provider (CSP) in these enterprises expand their data center capacities to accommodate this trend. The consumption of power in CSPs' data centers has increased 400% over the past decade [4]. In addition, some devices such as hard disks are the most vulnerable parts in such infrastructure, and the majority of hardware failure/replacement is due to hard disks [9]. Such infrastructure operation and maintenance would account considerable expenses. Even worse, data centers' carbon emissions continue to increase at a fast speed. Thus, it is important to consider the maintenance cost along with the power consumption of servers in order for CSPs to find a way to reduce the operational costs [7].

In this paper, a solution for infrastructure resource allocation in cloud computing system is proposed. The proposed methodology can be used to support the applications related to cloud computing, specifically in the infrastructure resource management, and to provide the capability of automatically allocating resources while taking the business objective such as load-balance of the utilization as well as the minimization of operational cost into account.

The rest of paper is organized as follows: Section 2 proposes the model for the infrastructure resource allocation problem and presents the idea of basic allocation strategy as the based methodology to be improved. Section 3 introduces an enhanced strategy to involving in cost saving consideration in the allocation process. Section 4 presents the

computational experiments for the purpose of comparing these two strategies. The paper is concluded with section 5.

2 Problem description

2.1 Model for business objective

In a typical enterprise cloud system, it needs the capability to automatically allocate resources by matching resource requirements upon the availability of resources, while taking both workload-balance and cost saving into account. Here is a sample for snapshot list of servers in cloud system data center [7]:

Table 1: the list of servers in cloud system

Server	CPU In Used	CPU In total	Memory In Used (G)	Memory In total (G)	Power Cost (cent/hour)	Maintenance Cost (cent/hour)
#0	1	4	1.0	8.0	3.5	2.1
#1	3	8	0.5	16.0	5.6	6.3
#2	2	16	2.0	12.0	7.5	4.5
# i	1	4	1.5	4.0	2.5	1.7
…	…	…			…	…

As shown in the list, a server's computational power is consist of CPU units and Memory size, and each server has its own operational cost when it is running. A general request to a CSP of a company, when a new task arrives the data center, the data center should allocate resources in order to meet the service requirement while taking the following business objectives into account: a) the load-balance, specifically the workload variance cross the servers in the data center should be as low as possible; b) cost saving, namely the cost sum of assigned servers should be kept lowest whenever it is possible. Here a weighted function may be introduced to make the trade-off of these two factors impacting the allocation decisions. Let define the weighted objective function to be $f = F_{min}(v, c, w)$, where v is for variance, c for cost, and w for the vector of weight values.

In the following, a model is proposed, which simplifies a bit the real problems encountered in the resource allocation for real cloud computing applications. Assume each server M_i whose workload could be summarized to a value O_i representing the occupancy of computing power by quantifying the workload of CPU, Memory and other devices with experience functions. The same principle may be applied to the cost consideration, where the resultant costs sue to the resource allocation could be represented by a value C_i. Let the indexes (i.e. i, j, m, n, k) stand for the identification of the server M. Suppose in a cloud system data center (E) contains k servers in total $(M_1, M_2, \dots M_k)$, and a candidate set (CS) contains some specified servers $(M_i, M_j \dots M_m, M_n)$, which is ready to assign task as the optimal result set to meet the incoming task requirements and business objectives. [3][5][7][8] The allocation process could be described as follows expression: based on a specified allocation strategy, sorting out a candidate set (CS) of the servers from a cloud system (E):

$$E(M_1, M_2, \dots M_k) \xrightarrow{allocation\ strategy} CS(M_i, M_j \dots M_m, M_n)$$

$$(0 \le i < j < m < n \le k) \quad (1)$$

2.2 Base Allocation Solution

The base allocation solution, which is also called Variance-Only Strategies, is a greedy algorithm designed to satisfy only variance-balance consideration depending on each server's occupancy value [11]. To minimize the variance of occupancy of the system using the principle of this "greedy algorithm", the BAS always chooses the server that has the least occupancy at the current step of allocation and assign task to it.

Define K to be the total number of servers in a data center with each server having occupancy of O_i, and let $\overline{n}$ be the average occupancy of all servers.

To meet the business goal for load-balance, the following objective function should be minimized:

$$S = \frac{1}{K}\sum_K^1 (O - \overline{n})^2 \text{ (2), where } \overline{n} = \frac{1}{K}\sum_{i=1}^{K} O_i$$

We can apply Lagrangian relaxation to generate an auxiliary function, and calculate the first-order partial derivatives of L that should be equal to zero in order to achieve an extreme value.

$$L = \sum_K^1 (o - \overline{n})^2 + \lambda\left(\frac{1}{K}\sum_{i=1}^{K} O_i - \overline{n}\right) \quad (3)$$

The solution is trial, and when $o = o_1 = o_2 = o_3 = \cdots = o_k = \overline{n}$, $\sum_K^1 (o - \overline{n})^2$ is minimal, and therefore, $S = \frac{1}{K}\sum_K^1 (o - \overline{n})^2$ is the optimal solution.

It is proved that the average value $\overline{n}$ is the key threshold value. Let $O = \{O_l, O_h\}$ be the set of occupancy of K servers that is divided into two components: O_l is subset of the occupancy is below the average value $\overline{n}$ while O_h is the one above $\overline{n}, O_l < \overline{n} < O_h$. Now assume that there is a request with R units to be assigned to servers evenly. Let the current variance be S, then after allocation the average value becomes: $\overline{n}' = \overline{n} + \frac{R}{K}$.

If all servers participate in allocation process, $S_1 = \frac{1}{K}\sum_K^1[(O + \frac{R}{K}) - \left(\overline{n} + \frac{R}{K}\right)]^2 = \frac{1}{K}\sum_K^1(O - \overline{n})^2 = S$.

If only O_l is involved in the allocation process, $S_2 = \frac{1}{K}[\sum_1^l(O_l' - \overline{n}')^2 + \sum_1^h(O_h - \overline{n}')^2]$ $(K = l + h)$. Now we can compare it with $S = \frac{1}{K}[\sum_l^1(O_l - \overline{n})^2 + \sum_h^1(O_h - \overline{n})]^2$, an additional constrain is $O_l' < \overline{n}' < O_h$, to ensure the consistency of the servers both subsets contained after the allocation.

$$f = \frac{(O_l' - \overline{n}')^2}{(O_l - \overline{n})^2} \sim \left|\frac{O_l + \frac{R}{l} - \left(\overline{n} + \frac{R}{K}\right)}{O_l - \overline{n}}\right| = \left|1 + \frac{R\left(\frac{1}{l} - \frac{1}{K}\right)}{O_l - \overline{n}}\right| \quad (4)$$

$o_l < \overline{n},\ l < K,\ R > 0 \rightarrow f < 1 \rightarrow \sum_l^1(o_l' - \overline{n'})^2 < \sum_l^1(o_l - \overline{n})^2$.

$$b = \frac{(O_h - \overline{n}')^2}{(O_h - \overline{n})^2} \sim \left|\frac{O_h - \overline{n}'}{O_h - \overline{n}}\right| \quad (5)$$

$O_h > \overline{n}' > \overline{n} > 0 \rightarrow b < 1 \rightarrow \sum_h^1(O_h - \overline{n}')^2 < \sum_h^1(O - \overline{n})^2 \ \therefore S_2 < S = S_1$.

Again assume the task requires r $(r > 0)$ resources. Let the occupancies of servers be $O = \{O_1, \dots, O_i, \dots, O_n, \dots, O_k\}$, which is sorted by occupancy value in ascending order. i.e., $(O_1 < \cdots < O_i \dots < O_n \dots < O_k)$, and O_n be the average value of the set. Following the derivation $S_2 < S_1$, the servers whose occupancy is below O_n are the favorable to be chosen for the allocation. To evaluate this allocation strategy, let variance S_3 be the one resulted from choosing the least-occupancy O_1 to assign task while S_4 be the variance resulted from choosing any other available server O_i:

$$S_3 = ([(O_1 + r)^2 + \dots + O_i^2 + \cdots + O_n^2 + \cdots + O_k^2])/K.$$

$$S_4 = ([O_1^2 + \cdots + (O_i + r)^2 + \cdots + O_n^2 + \cdots + O_k^2])/K.$$

$$S_3 - S_4 = \frac{2r}{K} * (O_1 - O_i) < 0.\ S_3 < S_4.$$

The conclusions drawn based on the above discussion are: a) the average occupancy value in the system $\overline{n}$ is the key threshold value. The server whose occupancy is less than average value is more favorable to be assigned to task, and b) the BAS with greedy algorithm idea could gain the optimal solution of minimizing workload variance by always assign task to the current least-occupancy server in once allocation process.

3 Cost optimization strategy

In this section, we are going to discuss the allocation strategy that takes the server operational cost factors into account. In a typical enterprise cloud computing system, the power cost and maintenance cost for a specified server could be considered as static attributions comparing to its constantly changing occupancy. In other words, the occupancy and cost for one server are two independent factors in this system. As a matter of factor, in order to consider workload balance and cost saving during the allocation process, the solution methodology needs to make a comprehensive evaluation for the 2-dimentional problem in the system, which in turn complicates the solution procedure. In order to avoid the compromise in the efficiency in solving the problem, a Cost Optimization Strategy (COS) is proposed that reuses the fundamental ideas of BAS and enhances the BAS by considering the operational cost during the allocation process. The proposed method reduces the problem's dimensionality by normalizing the cost and occupancy index. The normalized cost value as modification factor combining the actual operational cost and occupancy transfers a 2-dimentional problem into a 1-dimentional problem [5]:

$$P(cost, occupancy) \xrightarrow{modify\ occupancy\ by\ cost} P(occupancy')$$

3.1 Algorithm Steps

Supposing in a data center each server M_i has an instant occupancy O_i and the fixed cost C_i, where $O_i \in [O_{min}, O_{max}]$. We will organize these occupancy and cost values into 2 sets: $C = \{C_1, C_2, \dots, C_k\}$, $O = \{O_1, O_2, \dots O_k\}$. The algorithm can be described as the following steps:

1. Calculate normalize standard value Z_{oi} for $O_i \in O$ and Z_{ci} for $C_i \in C$, using normalization formula $Z = \frac{(\xi - \mu)}{\delta}$, where ξ stands for each individual value O_i or C_i, μ is average value, and δ is the unbiased variance for both set O and C. The result Z_{oi}, Z_{ci} could be added since they are dimensionless.

2. Calculate the modified normalized occupancy Z_{oi}', by using the amendment function with parameters Z_{ci} and weight factor w :$Z'_{oi} = Z_{ci} * w + Z_{oi}$ $(w > 0)$.

3. Restore the modified occupancy to a dimension unit by using linear transform $O_i' = Z_{oi}' * \delta_o + \mu_o$, here δ_o is the average value of set O and μ_o is the unbiased variance value of set O. Furthermore, the modified O_i' should be limited within the allowed range: if $O_i' > O_{max}$, set $O_i = O_{max}$, if $O_i' < O_{min}$, set $O_i = O_{min}$.

4. Use M_i' with modified occupancies O_i' instead of the M_i to join the allocation process, and allocate resources by using BAS discussed above.

When using COS, the process (1) could be update as:

$E(M'_1, M'_2, \dots M'_k) \overset{COS}{\Longrightarrow} CS(M'_r, M'_s \dots M'_x, M'_y)$ $(0 \le r < s < x < y \le k)$. (6)

The M' stands for server M with modified occupancy, and indexes (i.e.r, s, x, y, k) are for the identification of the server M'.

And the following conclusions are obvious:

If $C_i < \mu_c \rightarrow Z_{oi} < 0 \rightarrow Z_{oi}' < Z_{oi} \rightarrow O_i' < O_i$

If $C_i > \mu_c \rightarrow Z_{oi} > 0 \rightarrow Z_{oi}' > Z_{oi} \rightarrow O_i' > O_i$.

When the cost needs to be considered during the allocation process, based on the transformation we are able to see that a server with less cost would get a reduced modified occupancy value than its original one. According to the principle of BAS, the server with lower occupancy should has higher possibility to join the candidate set (CS), therefore, these low-cost servers could be likely picked to participate the allocation process. On the contrasts, a server with a higher cost would get an increased modified occupancy value, and it will be less likely picked for participating in the allocation procedure.

The weight w is designed as a bias ratio for the modified trade-off between occupancy consideration and cost consideration. The higher the weight w is, the bigger influence on cost side has. When w approaches to 0, the COS would be degenerated to BAS. In a typical enterprise system w is a variable that can be adjusted in different allocation processes to satisfy different business occasions and/or needs.

3.2 Enhancement

In COS, each server will update its actual occupancy O_i, and use O_i' to join in resource allocation process. If the solution method evaluates the occupancy index only, then some servers with actual higher occupancy could be assigned to process the requests from the task for cloud computing services that might cause variance deterioration. That means comparing to BAS, the variance after allocation would be getting worse by using COS. The detail data is demonstrated on the section of experiment.

To overcome this potential problem, the measure on generating O_i' phase should be improved. Since the average occupancy value $\overline{n}$ is the key threshold value, in the step of restoring occupancy O_i' we will use $\overline{n}$ as the upper/lower limit of the allowance range. Let $\overline{n} = \mathrm{O}_{\mathrm{mid}}$, we set the feasible range of O_i' to be $[O_{min}, O_{mid}]$ if the actual value $O_i \le O_{mid}$, and set the feasible range of O_i' to be $[O_{mid}, O_{max}]$ if the actual value $O_i > O_{mid}$. This extra constraint may avoid O_i' seriously departing from the actual value O_i by modifying itself with a great or small cost value with the goal of reducing the degree of variance deterioration.

4 Experiment

Using Monte Carlo method we created a set of data to simulate the scenarios of allocating resources in a cloud computing environment, and these corresponding problems were solved by BAS and COS respectively. The initial status (occupancy) for each server in the experiment was selected randomly between 0 and 100 and the operational cost is randomly distributed between 1 and 5.

Two experiments have been carried out. Experiment A shows how the operational costs are taken care during the allocation process by either BAS or COS. Experiment B presents the comparison of results (cost vs. workload balance) achieved by these two strategies. In order to fairly evaluate the efficiencies of the strategies in the computational experiments, the results are drawn from the average value derived from the results obtained based on 30-time repeated executions of the allocation process. The main parameters used in the computational experiment are:

R: request-total resource ratio, which indicates the rate relationship between the amounts of resources for which one single incoming cloud computing service task requests and the whole system own. E.g. 1:100 means the resource quantity required by a task is the one-hundredth of the whole system has.

w: weight used for the cost modification. When COS is employed, at the step on superposition of normalized values, the modified normalized value of occupancy of each server is determined by adding its normalized cost value multiplied by the specified weight value.

4.1 Experiment A

In this experiment, we investigate the results that are impacted by different R, where w in the experiment is fixed as 0.5.

Table 2: result in experiment A

R	1:100	1:500	1:1000	1:2000
BAS Cost Sum	81.23	101.02	103.67	112.99
COS Cost Sum	68.65	82.20	80.02	88.62
Cost Improvement	15.48%	18.63%	22.81%	21.57%

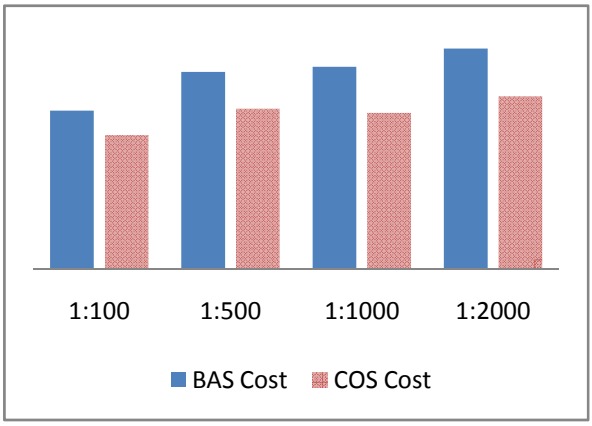

Fig. 1: result in experiment A in bar chart

The result shown in the bar chart demonstrates that comparing to BAS algorithm, COS is able to achieve the allocations with lower operational costs. Based upon the listed result, COS can achieve superior outcomes for all values of parameter R.

4.2 Experiment B

In this experiment, we generate two identical systems that will be used for the computational experiments of COS and BAS respectively. We choose different w values to investigate the efficiency of COS. We compare the results obtained by COS and BAS based on the following characteristics:

a). Percent of Cost Improvement

b). Percent of Variance Deterioration

c). Percent of Cost Improvement with varied weight: the value from a) multiplied by the specified weight w.

The following computational experiment results comes from the execution with R = 1:100.

Table 3: result in experiment B, R = 1:100, unit: %

W	0.1	0.25	0.5	0.75	1
Cost Improvement	3.22	11.04	16.89	24.33	37.20
Variance Deterioration	0.21	1.20	4.13	7.89	11.80
Cost Improvement *Weight	0.32	2.76	8.45	18.25	37.20

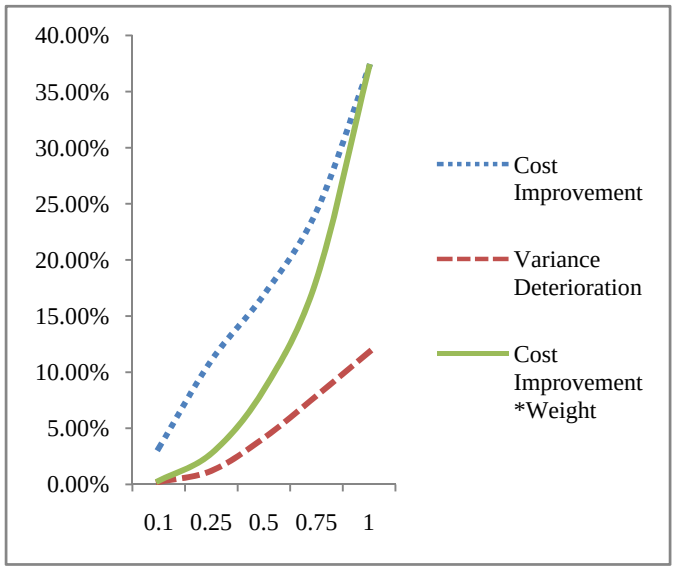

Fig. 2: result in experiment B in line chart, R = 1:100

The following computational experiments result comes from the execution R = 1:1000.

Table 4: result in experiment B, R = 1:1000, unit: %

Weight	0.1	0.25	0.5	0.75	1
Cost Improvement	6.52	11.34	26.23	32.66	45.01
Variance Deterioration	0.03	0.16	0.60	1.03	1.47
Cost Improvement *Weight	0.65	2.84	13.12	24.50	45.01

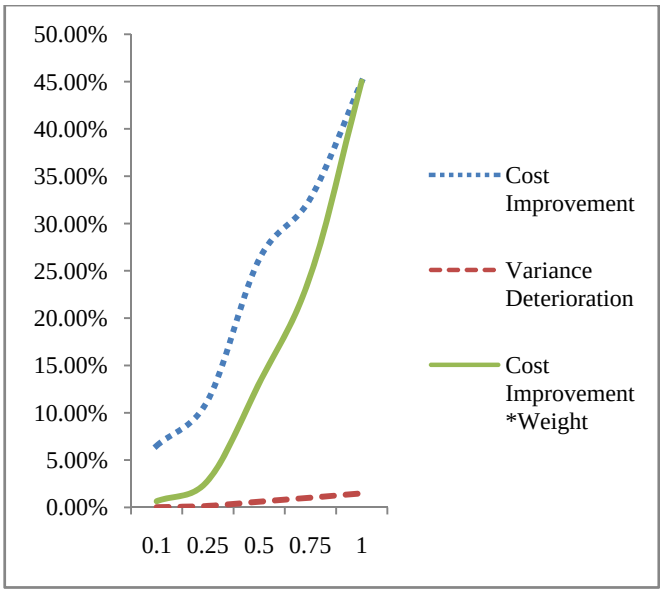

Fig. 3: result in experiment B in line chart, R = 1:1000

According to the results shown in the tables and the line charts we can find that the line for Cost Improvement × Weight is always above on the line for Variance Deterioration. The outcomes demonstrate the following conclusions: a) The advantages on cost reduction is far more obvious than the disadvantages on variance change via COS. b) the efficiency of COS is directly proportional to w ($w \in (0,1]$), and inversely proportional to R.

Based on the results of all computational experiment listed above and other computational experiments we conducted, the Cost Optimization Strategy (COS) is proven to be able to effectively make the tradeoff decisions during the resource allocation to consider both workload-balance and cost saving. In the real work applications, an enterprise cloud computing system has vast computing power and huge amount of servers, the request-total resource ration *R* would be a tiny value, therefore, in actually practice, the COS will yield more satisfactory outcomes.

5 Conclusions

In this paper, we present first a model to address the problem of infrastructure resource allocation in cloud computing systems in order to provide the service of automatically allocating resources. The allocation procedure takes the overall workload-balance and allocation costs into account. After the model is introduced, we propose the solution methodology, Cost Optimization Strategy, aims enhancing Base Allocation Strategy to consider the allocation costs in addition to the workload balance. Computational experiments are conducted to validate the proposed methods. Based on the experiment results, the Cost Optimization Strategy is proven to be capable to make allocation decisions effectively by considering both workload balance and allocation costs when a cloud computing request is processed.

6 References

[1]. The NIST Definition of Cloud Computing, *National Institute of Standards and Technology, U.S. Department of Commerce*

[2]. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar A. F. De Rose,Rajkumar Buyya. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms

[3]. Eddy Caron, Frederic Desprez, David Loureiro. Cloud Computing Resource Management through a Grid Middleware: A Case Study with DIET and Euca-lyptus, *2009 IEEE International Conference on Cloud Computing*

[4]. Filani,D., He, J., GAO, S., RAJARPA, M., KUMAR, A., SHAN, R.AND NAAPPAN R. Dynamic data center power management: Trends, issues and solutions.

[5]. GANAPATHI A., CHEN Y., FOX A., KATZ R. AND PATTERSON D. Statistics-Driven Workload Modeling for the Cloud

[6]. Ali Khajeh-Hosseini, Ian Sommerville, Ilango Sriram. Research Challenges for Enterprise Cloud Computing

[7]. Haiyang Qian, Deep Medhi, Server Operational Cost Optimization for Cloud Computing Service Providers over a Time Horizon

[8]. TANG Xiao-chun, LIU Jian. A research on cloud infrastructure allocation algorithm based on meta-zone

[9]. VISHWANATH, K.V., AND NAGAPPAN, N. Characterizing cloud computing hardware reliability. *In Proc. Of 1st ACM Symposium on Cloud Computing (June 2010).*

[10]. Lizhe Wang, Jie Tao, Marcel Kunze, Scientific Cloud Computing: Early Definition and Experience, The 10th IEEE International Conference on High Performance Computing and Communications

[11]. Guiyi Wei, Athanasios V. Vasilakos, Yao Zheng, Naixue Xiong. A game-theoretic method of fair resource allocation for cloud computing services