ARAS-M: Automatic Resource Allocation Strategy based on Market Mechanism in Cloud Computing

Xindong You
School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou , China.
youxindong@hdu.edu.cn,

Jian Wan, Xianghua Xu, Congfeng Jiang, Wei Zhang, Jilin Zhang
School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou , China.
{wanjian@hdu.edu.cn, xuxhcs@zju.edu.cn}

***Abstract*—Resource management is one of the main issues in Cloud Computing. In order to improve resource utilization of large Data Centers while delivering services with higher QoS to Cloud Clients, an automatic resource allocation strategy based on market Mechanism (ARAS-M) is proposed. Firstly, the architecture and the market model of ARAS-M are constructed, in which a QoS-refectitive utility function is designed according to different resource requirements of Cloud Client. The equilibrium state of ARAS-M is defined and the proof of its optimality is given. Secondly, A Genetic Algorithm (GA)-based automatic price adjusting algorithm is introduced to deal with the problem of achieving the equilibrium state of ARAS-M. Finally, ARAS-M is implemented on Xen. Experiment results show that ARAS-M can approximately achieve the equilibrium state, that is, demand and supply is nearly balanced, which validates that ARAS-M is effective and practicable, and is capable of achieving resource balance in cloud computing.**

***Index Terms*—Cloud Computing; Resource Allocation; Market Mechanism; Genetic Algorithm**

I. INTRODUCTION

The latest computing paradigm to emerge is Cloud Computing [1], which promises reliable services delivered through next-generation data centers that are built on computing and storage virtualisation technologies. A number of computing researchers and practitioners have attempted to define Clouds in various ways [2]. For example Buyya [3] define Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers. Virtualized technologies are the main feature of Cloud systems. Employing the virtualized technologies, such as VMWare PC, VMWare ESX, Xen, and KVM etc, one or more VMs (Virtual Machines) can run on a physical machine simultaneously. Therefore, resource management in Cloud Computing systems is at a finer granularity (at VM layer) and is more agile. On the other hand, one of the most important objectives of Cloud system is to provide high quality and transparent services to Cloud clients through improving the utilization of the large Data Centers. However, the available resources of the providers and the resources requirement of consumers are both changing dynamically. Hence, how to manage resources dynamically and agilely in terms of the varied requirements of consumers is a challenge in Cloud Computing environments. In this paper, an automatic resource allocation strategy based on market mechanism (ARAS-M) is proposed to address this problem. In the proposed strategy, Equilibrium Theory is introduced, the equilibrium state is defined and its optimality is proved. ARAS-M will try to achieve the equilibrium state through employing the present GA (Genetic Algorithm)-based automatic price adjusting algorithm. Preliminary experiment results on Xen validate that ARAS-M is practicable and effective.

The rest of this paper is organized as follows: Related work on resource management based on economy theory is investigated in Section2. Detailed description of our proposed resource allocation strategy ARAS-M appears in Section3. The present GA-based automatic price adjusting algorithm is described in Section 4. And the experiments and presents the preliminary results are demonstrated in Section 5. Finally, we conclude the paper with future work in Section6.

II. RELATED WORK

Employing economy theory to manage resources especially to balance loads among the distributed systems has received extensive researches in the past twenty years. There are many published papers and prototypes on it [4,5,6], of which the main idea is using the signal of price to reflect resource utilization. By means of allocating new jobs to the cheapest resource achieves

doi:10.4304/jcp.6.7.1287-1296

loads balanced among distributed resources. Economic theories are also widely used in Grid Computing to solve the problem of providing dynamically available resources to the varying requirements of resource consumers. Rajkumar Buyya firstly introduces the market mechanism to manage Grid resources [7]. A grid simulator called GridSim is built to simulate the behavior of market entity, and the desired objectives are obtained. Cao et al propose a market-based approach to allocate resources for Computational Grids [8], in which General Equilibrium Theory is employed to maximize profit of the resource consumers and providers while market mechanism is adopted to balance the resource requirements and supplies. Jiang et al study and analyze the different price adjustment strategies. With MAS (multi agent system) coordinated technology and market bidding game rules, a grid resource allocation model based on market economy is introduced in [9], which makes the allocation of the whole network resource tend to be more reasonable. The research group led by Jordi Guitart are focus on how to use an Economically Enhanced Resource Manager to maximize revenue in Grid Markets. And their research results are published in some journals or conferences [25,26,27,28,29].

All above related researches show that market economy is suitable for solving resource management problem and has its advantages. However, their research results can't be employed in Cloud Computing environments directly. Resource management in Clouds is at a finer granularity and in more levels. At the lowest level, VMM (Virtual Machine Monitor) built on physical machine is responsible to allocate fractions of CPU, memory, disk and network to the different VMs, who are installed upon the VMM. At the middle level, Cluster Manager manages all the VMs in the cluster: allocate different VMs to different applications based on different management strategies, and VM can be migrated among different physical machine to achieve workload balanced. While at the top level, Clouds Manager should determine how to select appropriate cluster or PC to run different applications of Cloud Clients with their QoS met. Resource management strategies based on market economy theories should be modified due to distinctive features of Cloud Computing. Buyya prospects market-oriented Cloud Computing, describes the vision, hype, and reality for delivering IT Services as Computing Utilites [3]. In this paper, we firstly apply the market economy mechanism to manage Cloud resources in the lowest level (VMM level). It is easily extended the resource strategy to the upper level that is Cluster level in future.

On the other hand, Autonomic computing systems can regulate and maintain themselves without human intervention. Such systems are able to adapt to changing environments (such as changes in the workload or failure) in a way that preserves given operational goals (e.g., performance goals). There has been significant research and attention to autonomic computing in the recent years [14,15,16,17,18]. Recently, applying the autonomic controller theory to resource allocation in virtual environments is also received research interests [19,20,21,22,23,24].

In this paper, we also firstly combine the controller theory and market economy theory to solve the dynamic resource allocation problem in cloud computing environments.

III. AUTONOMIC RESOURCE ALLOCATION STRATEGIES BASED ON MARKET MECHANISM ARAS-M

Resource allocation in current VMM such as VMWare, Xen, KVM, et al, is through schedulers implemented in the hypervisor [10]. Although there are schedulers modified to meet requirements of different type applications [11,12], they all preset one of the schedulers and allocate resource statically. Static resource allocation mechanism is inefficient to meet the dynamic resource requirements of consumers, and it is disadvantage to increase resource utilization. Our proposed resource allocation strategy described in the following subsections dynamically allocates resource fractions according to varying resource requirements, which can improve resource utilization while maximizing benefits of both service providers and resource consumers at the same time through employing the market mechanism.

A. Architecture of ARAS-M

VMM with high performance is usually built upon the physical machine directly, and have control of underlying physical resource. Based on a certain strategy, fractions of resources are allocated by VMM. Architecture of our market-based resource allocation strategy ARAS-M is shown in Figure 1.

Workload submitted by Cloud consumer will run on a VM ultimately. The number of fractions can be used by one VM is determined by ARAS-M, and can be adjusted dynamically according to the varied resource requirement of workload. Architecture of ARAS-M mainly consists of three parts: Consumer Agent (CA), Resource Agent (RA) and Market Economy Mechanism. Consumer Agent (CA) delegates the consumer to participate in the market system and aims to obtain maximal benefit for the consumer. RA delegates one type of resource to publish the resource's price and to adjust the price according to the relationship of supply and demand in the market system. Market Economy Mechanism is responsible for balance the resource supply and the demand. On the whole the aim of ARAS-M is to maximize the profits of both CA and RA by means of balancing the demand and supply in the market, and to improve the resource utilization ultimately.

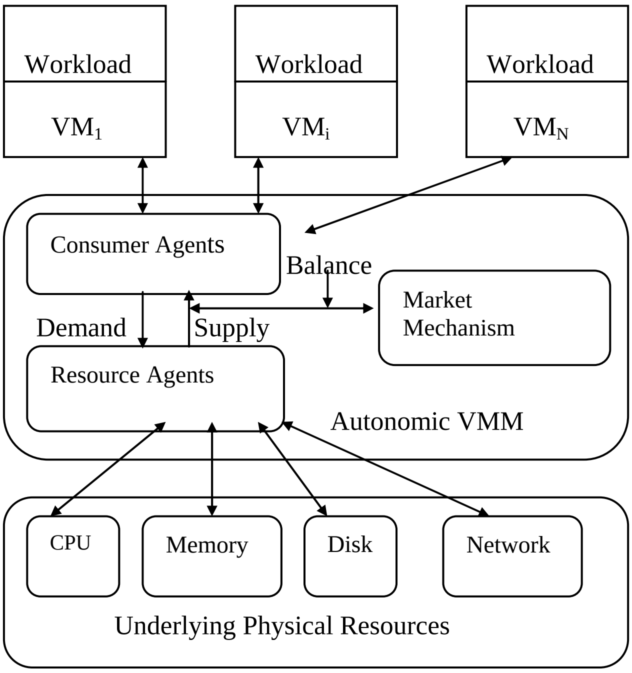

Figure 1. Architecture of ARAS-M

B. Market Model of ARAS-M

According to the architecture of ARAS-M, Market Model of ARAS-M mainly consists of the following three parts:

① $W = \{w_1, w_2, \cdots, w_n\}$ is the set of workload submitted by Cloud clients whose size is n;

② $R = \{r_1, r_2, \cdots, r_m\}$ denotes the type of resources controlled by VMM , and the number is m;

③ $P = \{p_1, p_2, \cdots, p_m\}$ is a price vector whose size is m, in which p_j represents the price of r_j when all capacity used.

Based on the above definition, the problem of resource allocation in Cloud Computing is transformed into the process of finding out an $n \times m$ size matrix f , in which f_{ij} denotes fraction of resource r_j allocated to the workload w_i running on a VM. Hence, the feasible solution set is defined as bellow:

$$F = \{ f \mid 0 \le f_{ij} \le 1, 0 \le \sum_{i=1}^{n} f_{ij} \le 1 \}.$$

Transforming matrix f into a row vector, that is $f = (f_1, f_2, \cdots f_n)^T$, the vector $f_i = (f_{i1}, f_{i2}, \cdots, f_{im})$ denotes fractions of all type resources allocated to the workload w_i .

In order to analyze or implement ARAS-M conveniently, we classify workloads that are running on VM. Recently, workloads running on VMs can be approximately classified into four types: computing-intensive, data-intensive, network I/O-intensive and latency-sensitive. The relationship among different workloads and different resources is different, and leads to different performance. A matrix C with $n \times m$ is constructed to denote their relationships, where c_{ij} denotes the correlation coefficient of workload w_i and resource r_j . Cloud client's satisfaction with service is related to the fractions of different type resources allocated to him. Hence, utility function $u_i(f_i) = u_i(f_{i1}, f_{i2}, \cdots, f_{im})$ is introduced to represent the degree of the client's satisfaction. The utility function usually has the following properties:

① $u_i(f_i)$ is concave, that is, if $u_i(f_i^1) = u_i(f_i^2) = U$, for any $t \in 0 \le t \le 1$, $u_i(tf_i^1 + (1-t) f_i^2) \ge U$ is always true;

② $u_i(f_i)$ is monotone increasing, that is for any $j \in \{1,2 \cdots m\}$, $f_i^1 \ge f_i^2$, and if there exist one $j \in \{1,2 \cdots m\}$ making $f_i^1 > f_i^2$, then $u_i(f_i^1) \ge u_i(f_i^2)$ is always true.

The concave property of utility function $u_i(f_i)$ indicates QoS cannot be improved remarkablely through only increasing the allocation fraction of one type resources, and the improved QoS related with increased allocated fractions of more than one type resources. Monotone increasing property of utility function $u_i(f_i)$ means more fractions allocation for one client can make its QoS improved at a greater degree. We design the utility functions who have the above two properties:

$$u_i(f_i) = k_1 f_{i1}^{c_{i1}} + k_2 f_{i1}^{c_{i1}} + \cdots + k_m f_{im}^{c_{im}} \quad (1)$$

When $0 \le c_{ij} \le 1$, we can prove the designed utility functions are monotone increasing and are concave.

Define $B_i(P)$ is the maximal benefit received by workload w_i under the price vector of P. That is:

$$B_i(P) = \max_{f_i \in [0,1]^m} [u_i(f_i) - P f_i^T] \quad (2)$$

And the total utility of all CAs can be defined as bellow when the fraction resource allocation is f :

$$W(f) = \sum_{i=1}^{n} u_i(f_i) \quad (3)$$

If the fraction solution f maximizes $W(f)$, we say the fraction solution f is an optimal solution.

The process of allocation reaches the equilibrium state only if the solution f and price vector P satisfy the following three conditions:

① $f \in F$;

② for any w_i , $u_i(f_i) - P f_i^T = B_i(P)$ that is, any CA receives his maximal benefit;

③ for all r_j, $\sum_{i=1}^{n} f_{ij} = 1$, that is, capacity of resource r_j is fully used, no fractions are wasted.

Explanation for the above conditions is that if f is an equilibrium solution, the solution f has the following three properties:

(1) The solution f is feasible;

(2) Every CA gains his maximal benefit when the price vector is P according to the allocation fractions in f;

(3) Every type of resource in the system is fully used and no fractions are wasted.

The relation of the optimal solution and the equilibrium solution can be described as **Theorem 1**.

Theorem 1: When coming to the problem of resource allocation in Cloud Computing environment, if a solution f is an equilibrium solution when price vector is P, then the solution f is optimal naturally.

Proof: ∵ if f is an equilibrium solution when price vector is P, then $u_i(f_i) - Pf_i^T = B_i(P)$ is true.

$$\Rightarrow u_i(f_i) = Pf_i^T + B_i(P)$$

$$W(f) = \sum_{i=1}^{n} u_i(f_i) = \sum_{i=1}^{n} Pf_i^T + \sum_{i=1}^{n} B_i(P)$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m} p_j \cdot f_{ij} + \sum_{i=1}^{n} B_i(P)$$

$$= \sum_{j=1}^{m} p_j \sum_{i=1}^{n} f_{ij} + \sum_{i=1}^{n} B_i(P)$$

∵ For all r_j has $\sum_{i=1}^{n} f_{ij} = 1$ when the solution f is equilibrium (according to condition ③).

$$\therefore \quad W(f) = \sum_{j=1}^{m} p_j + \sum_{i=1}^{n} B_i(P)$$

∵ The price vector in one equilibrium state is fixed.∴ The value of $\sum_{j=1}^{m} p_j$ is a constant. According to the definition of $B_i(P)$ (formula 2), $B_i(P)$ is the maximal benefit received by CA$_i$. Therefore, there will be no larger value than $W(f) = \sum_{j=1}^{m} p_j + \sum_{i=1}^{n} B_i(P)$. That is, $W(f)$ is the maximum, so the fractions vector f is the optimal solution.

Proposition is proved.

Theorem 1 shows that if solution f is an equilibrium solution when price vector is P, the solution is optimal one also. Under the equilibrium state, the benefits of all CAs and RAs all reach their peak. That is to say the solution f is the most reasonable allocation.

On the other hand, according to the general equilibrium theory [13], if the utility function satisfies the conditions of concave and monotone increasing, the existence of the equilibrium solution is natural.

Therefore, the problem of resource allocation in Cloud Computing is transformed into a process of finding out the equilibrium price vector and the equilibrium solution, which will be described in the next section in detail.

IV. GA-BASED AUTOMATIC PRICE ADJUSTING ALGORITHM

From the above description, we know that the problem of resource allocation based on market economy theory is transformed into finding out the equilibrium price vector and equilibrium solution corresponding. In this paper, a GA-based (Genetic Algorithm based) automatic price adjusting algorithm is proposed. For any resource r_j ($j \in \{1,2\cdots m\}$, initially set his price respectively. To maximize profits, CAs submit their demand fractions to all types of resource according to their utility function. After all fractions collected, every type resource r_j adjusts his price according to the relationship between the demand and the supply. Price will be automatically controlled until the demand and the supply balanced. The control flow diagram is shown as Figure 2.

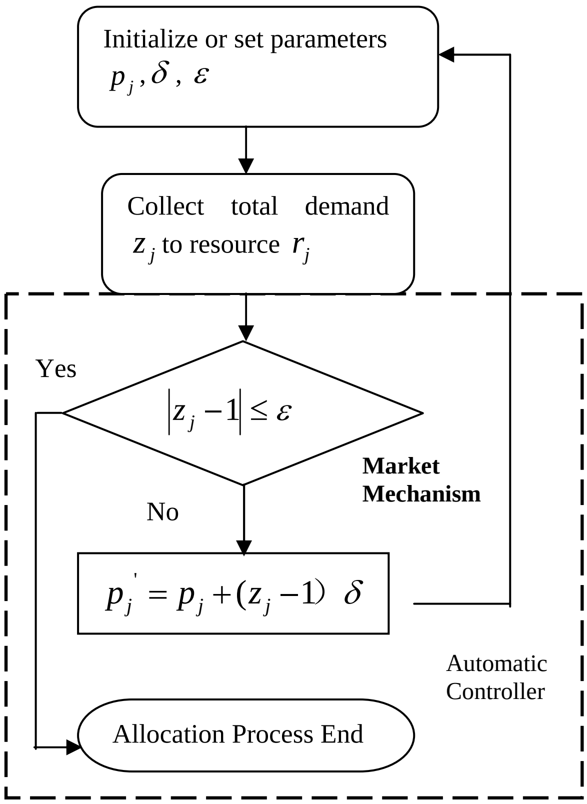

Figure 2. Automatic Controller Frame

The details of our proposed GA-based automatically price adjusting algorithm is described below:

① According to the type of workload submitted by CAs, their correlation coefficients is denoted by a $n \times m$ size matrix C;

For any resource r_j ($j \in (1 \sim m)$), the following operations will be executed asynchronously:

② Resource Agent of r_j set his initial price p_j, the rate of price adjusting δ, and the terminal parameter ε;

③ After the price vector is set, a Genetic Algorithm is utilized to find the optimal solution f, which makes the CA achieve his maximal benefit;

④ Under the optimal solution f, calculating the total demand z_j to resource r_j of all of CAs;

⑤ If $\left|z_j - 1\right| \le \varepsilon$, the current price p_j is the equilibrium price, the corresponding fractions of resource allocation f is the equilibrium solution, resource r_j allocates the fractions to the CAs according to the solution f, resource allocation of r_j ends. Return to ①; Else, execute the following operations continue:

⑥ Resource Agent adjusts the price based on the formula: $p_j^{'} = p_j + (z_j - 1)\ \delta$, then the new price $p_j^{'}$ is got;

⑦ If $p_j^{'}$ is negative, adjusting the rate of price adjusting according the formula: $\delta = \frac{\delta}{2}$, once the new rate of price adjusting is got, return to⑥;

⑧ Based on the new price $p_j^{'}$, repeating operate ②～⑦ until the approximate equilibrium solution is got, that is $\left|z_j - 1\right| \le \varepsilon$, the demand and the supply of resource r_j is balanced approximately.

The main idea of our proposed GA-based automatic price adjusting algorithm is: Utilizing principle of price lever in market mechanism, when the demand of resource r_j exceeds its supply, the price of resource r_j will be raised at a certain rate, otherwise, will be dropped. Price adjusting is automatically based on market mechanism, and the system of our market will reach its equilibrium state finally. On the other hand, Genetic Algorithm induced guarantees CA get his maximal benefit approximately. After a certain number of times, the market system will be in an equilibrium state, in which both Consumer Agents and Resource Agents get their maximal benefit.

V. EXPERIMENTS AND RESULTS

In order to validate the efficiency and feasibility of our proposed GA-based automatically price adjusting algorithm, we conduct our experiments on Xen, upon which four VMs are installed and four types of workloads submitted by CA run on the VMs respectively. The utility function of CA is also submitted, and the algorithm of resource requirement adjusting according to its utility function is implemented in the CA. The automatically price adjusting algorithm is implemented in RA. RA is running on VMM (Virtual Machine Monitor) while CA is running on VM. The interactive operations between CA and RA will accomplish the task of resource allocation. The experiments parameters are described as below:

The correlation coefficient vector C ＝ [0.1,0.2,0.3,0.4], and weight parameters of the utility function is $k_1 = k_2 = k_3 = k_4 = 1$. The procedures of GA, coding method of GA, fitness function of GA, three operators of GA in our proposed GA-based automatically price adjusting algorithm, are described in detail as below:

P1: Producing the initial population $\{G^r\}_{r=1}^{pop\_size}$ with *pop_size* scale, (*pop_size*=20 in our experiments), every population G^r is denoted as a chromosome with gene coding. Coding method adopted in our experiments is float value with multi-parameters co-coded. The number of parameters is 4 in our experiment. Bounds of the parameter are [0,1], which represents the proportion of resource allocated to VMs;

P2: Designing the fitness function $F(f_i)$ to calculate the fitness of every population, the function adopted in our experiments is the benefit function: $F(f_i) = u_i(f_i) - Pf_i^T$;

P3: Designing Selection Operator, based on which the next generation populations are selected. The larger of value of the fitness of one population has more chance to be chosen to exist in the next generation. The NormGeomSelect Operator is adopted in our algorithm, in which the ranking selection function is based on the normalized geometric distribution.

P4: Designing Crossover Operator: Generate the next generation population according to a certain cross probability and cross method. The arithmetic Crossover Operator is adopted in our algorithms, which takes two parents and performs an interpolation along the line formed by the two parents;

P5: Designing Mutation Operator: Generate the next generation population based on a certain mutation probability and mutation method. The non-uniform Mutation Operator is adopted in our algorithm, which changes one of the parameters of the parent based on a non-uniform probability distribution;

P6: Generating the next generation population by the operation of Crossover and Mutation Operator;

Repeating do P2～P6 until satisfy the conditions of terminal.

Agent of CPU sets the different initial prices and more than one time of our experiments are carried out.

Experiments results obtained are described in the following figures and tables (where the CPU supply is 1).

After the equilibrium price vector is found, Genetic Algorithm will be utilized to search the optimal solution to maximize the benefit of CA. The search processes under the above condition (Figure 3) are shown in Figure 4~7.

The equilibrium price under the above situation is 147.3139, based on which the fraction of resource allocation to the four CAs and their maximal benefit are shown in Table2.

All the experiment results show that our proposed ARAS-M, through the GA-based automatically price adjusting algorithm can achieve the equilibrium state, that is, the demand and the supply in the market is balanced (all demand ≈ 1, margin of demand and supply ≈ 0). Moreover, when the initial set price is below the final equilibrium price, the price will be raised, and the demands (demand1~demand4) of all CAs shrink after running our proposed GA-based it automatically price adjusting algorithm (shown as Figure 3). When the initial set price is close to the equilibrium price, the resource price fluctuates near the equilibrium price, and reaches balanced approximately (shown as Figure 8). And when the initial price is higher than the equilibrium price, the resource price will drop, and the demand of all CAs will be enlarged, the balanced state will be also obtained (shown as Figure 9).

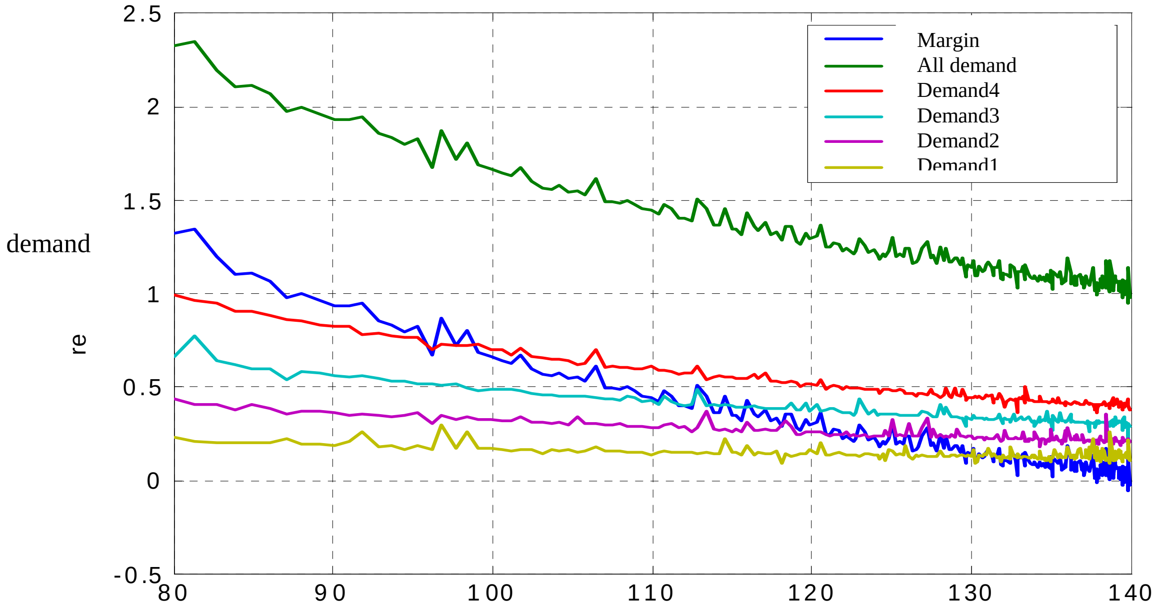

Figure 3. Demand varied with price when initial price $p_j = 80$, rate of price adjusting $\delta = 1.5$, terminal parameter $\varepsilon = 0.001$

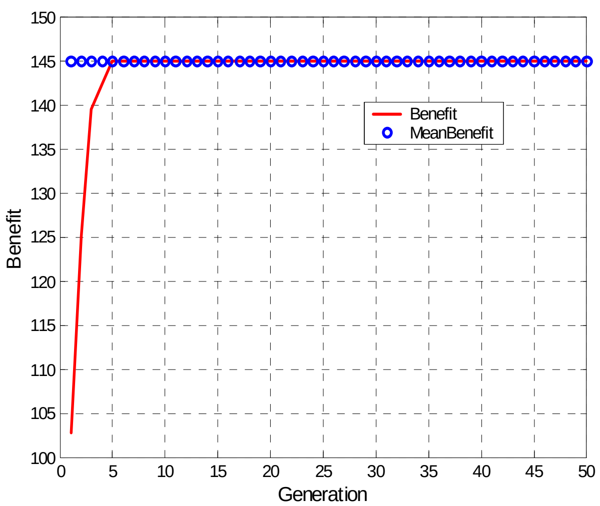

Figure 4. Maximal Benefit Searching Process of CA1
Final fraction= 0.1152; Best Benefit=145.0159

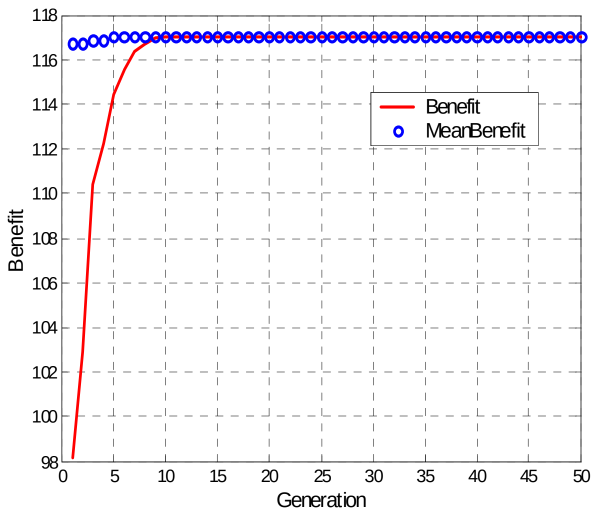

Figure5. Maximal Benefit Searching Process of CA2
Final fraction= 0.2094; Best Benefit= 117.0036

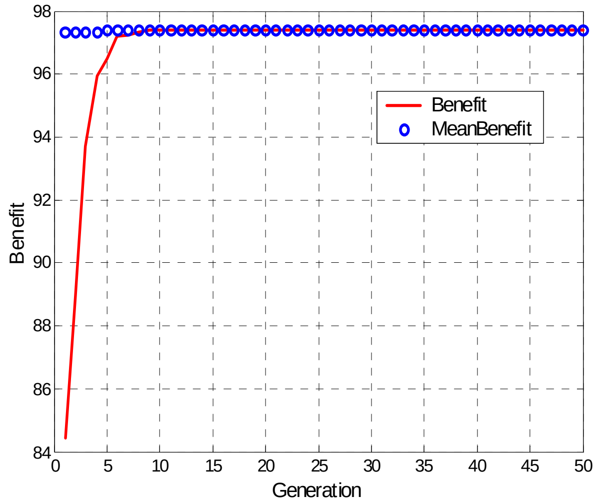

Figure6. Maximal Benefit Searching Process of CA3 Final fraction= 0.3077; Best Benefit= 97.3931

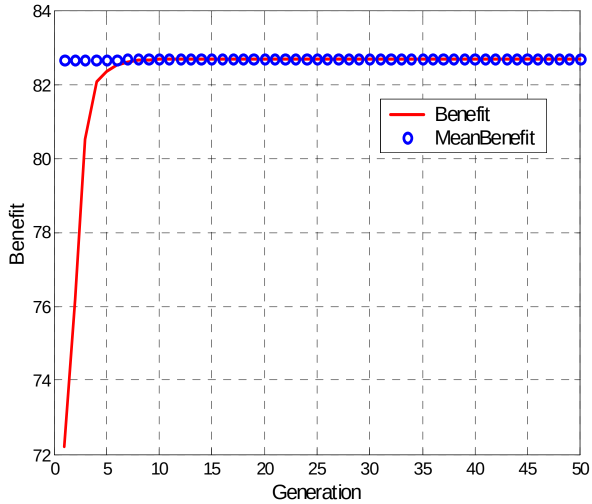

Figure7. Maximal Benefit Searching Process of CA4 Final fraction= 0.3942; Best Benefit= 82.6824

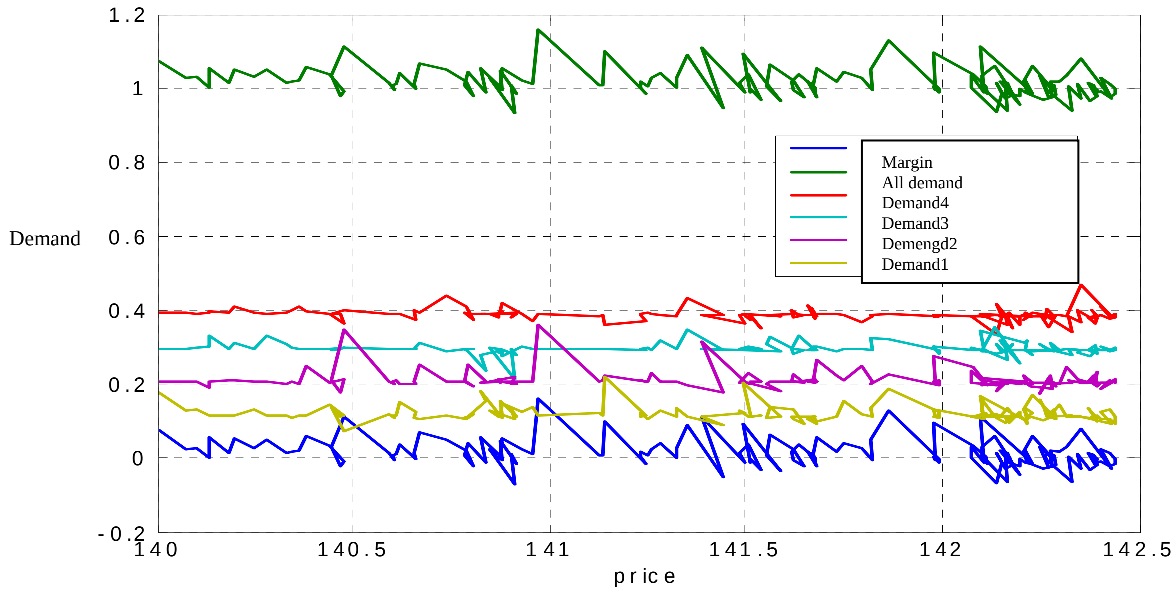

Figure 8 Demand varied with price when initial price $p_j = 140$, rate of price adjusting $\delta = 1$ terminal parameter $\varepsilon = 0.0001$

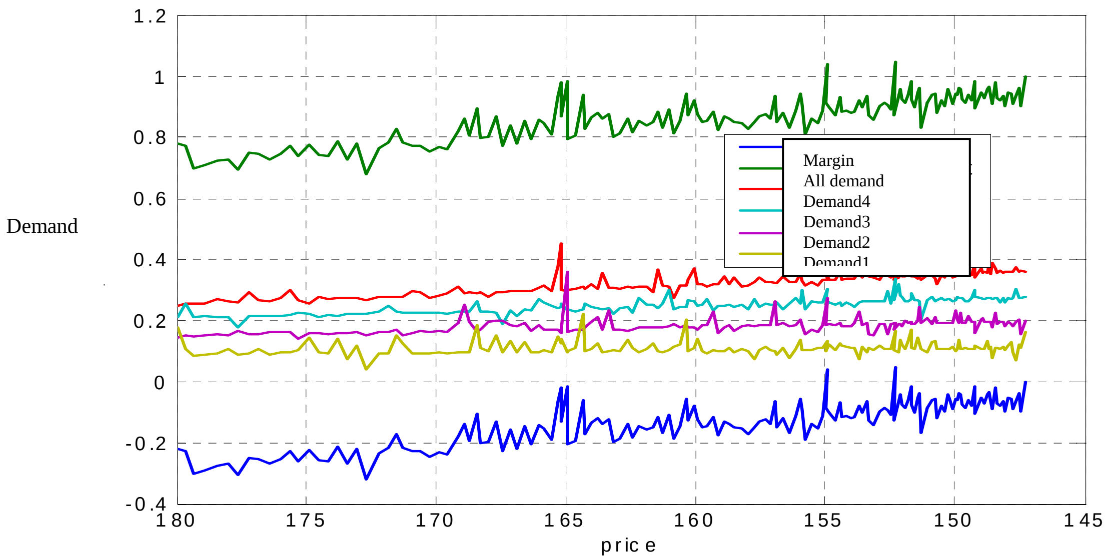

Figure 9 Demand varied with price when initial price $p_j = 180$,rate of price adjusting $\delta = 1.5$,terminal parameter $\varepsilon = 0.001$

TABLE I: FRACTIONS AND BENEFITS UNDER THE EQUILIBRIUM STATET

Equilibrium price =142.103	CA1	CA2	CA3	CA4	Total
Fractions	0.1132	0.2100	0.2918	0.3850	1.000
Benefit of CAs	144.7615	116.5356	96.7494	81.8160	439.8625

TABLE II: FRACTIONS AND BENEFITS UNDER THE EQUILIBIRUM STATE

Equilibrium price =147.3139	CA1	CA2	CA3	CA4	Total
Fractions	0.1651	0.1969	0.2772	0.3599	0.9990
Benefit of Cas	142.7123	115.4977	95.2677	79.8749	433.3526

Under the equilibrium price vector, Genetic Algorithm is employed to find out the solution fraction with maximizing the benefit of the CA respectively. The equilibrium price and the equilibrium solution obtained from the three experiments are shown in the Table1 and Table2 respectively.

Our experiment results show that our proposed algorithm in our ARAS-M framework under the Cloud Computing environment can improve the resource utilization while maximizing the benefits of all CAs. Both the consumers and providers of Cloud service gain their maximal profit, which validates our ARAS-M is practicable and feasible.

VI. CONSULION AND FUTURE WORK

Cloud computing is a new emergent computing paradigm, in which resource management is one of the most important parts. The advantages of employing market economy mechanism to solve the problem of resource allocating in Cloud Computing environment with property of dynamics are studied and analyzed in this paper. A market-based resource allocation strategy ARAS-M in Cloud Computing is proposed: Firstly, utility functions of CAs are constructed to denote their satisfaction with fractions allocated to them, then the equilibrium state is defined and its optimality is proved, finally a GA-based automatically price adjusting algorithm is present to deal with the problem of balancing the demand and supply in our market model. Experiment results obtain from Xen validate our ARAS-M is effective and practicable.

Currently, our market-based resource allocation strategy ARAS-M is only implemented to allocate resource in the lowest level of Cloud Computing, and only manage the CPU resource. In the future, implementing our ARAS-M in the upper level resource management module of Cloud Computing and extending to manage more types of resources will be explored.

ACKNOWLEDGMENT

This work is supported by State Key Development Program of Basic Research of China (Grant No. 2007CB310900) and Natural Science Foundation of China (Grant No.60873023, 60973029, 61003077) Natural Science Foundation of Zhejiang Province (Y1101092,Y1101104, Y1090940),Startup Foundation of Hangzhou Dianzi University KYS05560803,

REFERENCES

[1] A. Weiss. Computing in the Clouds. NetWorker, 11(4):16-25, Dec. 2007.

[2] Twenty Experts Define Cloud Computing, http://cloudcomputing.syscon.com/read/612375_p.htm [18 July 2008].

[3] R.Buyya, S.Yeo Chee, S.Venuqopal. Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Servers as Computing Utilities[A]. *Cloud IT Platforms 2008* [C]. San Diego: Springer Press. 2008: 5~13.

[4] Sermret Nemo. *Market mechanisms for network resource sharing*[D]. New York: Coblumbia University, 1999.

[5] Nathaniel R B. *Economic allocation of computation time with computation market*[D]. Cambridge: Massachusetts Inst Tech, 1994.

[6] Lalis S, Karipidis A. An open market-based architecture for distributed computing [A]. *In IPDPS workshop 2000* [C]. Cancum: Springer Press. 2000: 61-70.

[7] Buyya R, Vazhkudai S. Compute Power Market: Towards a Markte-Oriented Grid[A]. CCGRID 2001[C]. Brisbane: IEEE Press. 2001: 571-581.

[8] CAO Hong-Qiang, XIAO Nong ,et al. A Market-based Approach to Allocation Resources For Computational Grids[J]. Journal of Computer Research and Development, 2002,39(8): 913-916.

[9] JIANG Wei-Jin, WANG Pu. Research on a Grid Resource Allocation Algorithm Based on MAS Non-Cooperative Bidding Game [J]. Journal of Computer Research and Development, 2007,44(1): 29~36

[10] Credit scheduler. http://wiki.xensource.com/xenwiki/CreditScheduler.

[11] Govindan S, Nath Arjun R, Das A, et al. Xen and Co.: Communication-aware CPU Scheduling for Consolidated Xen-based Hosting Platforms [A]. VEE07[C]. San Diego: ACM press. 2007: 126-136.

[12] Ongaro D, Cox Alan L, Rixner S. Scheduling I/O in Virtual Machine Monitors [A]. VEE0 [C]. Seattle: ACM Press. 2008: 113-122.

[13] Varian Hal R. Microeconomic Analysis[M].New York: W W Norton & Company, 1992.

[14] Babaoglu, O., Jelasity, M., Montresor, A.: Grassroots Approach to Self-Management in Large-Scale Distributed Systems. In Proc. EU-NSF Strategic Research Workshop

on Unconventional Programming Paradigms, Mont Saint-Michel, France, 15-17 September (2004)
[15] M.N. Bennani and D.A. Menasc´e, "Assessing the Robustness of Self-managing Computer Systems under Variable Workloads, Proc. IEEE Intl. Conf. Autonomic Computing (ICAC'04), New York, NY, May 17–18, 2004.
[16] D. A. Menasc´e and M.N. Bennani, "On the Use of Performance Models to Design Self-Managing Computer Systems," Proc. 2003 Computer Measurement Group Conf., Dallas, TX, Dec. 7-12, 2003.
[17] D. A. Menasc´e, "Automatic QoS Control," IEEE Internet Computing, Jan./Febr. 2003, Vol. 7, No. 1.
[18] D. A. Menasc´e, R. Dodge, and D. Barbar´a, "Preserving QoS of E-commerce Sites through Self-Tuning: A Performance Model Approach," Proc. 2001 ACM Conf. Ecommerce,Tampa, FL, Oct. 14-17, 2001
[19] D. A. Menasce, and N Bennani. " Autonomic Virtual Environments". In proc. International Conference on Autonomous Systems (ICAS), Silicon Valley, CA. July 19~21, 2006.
[20] P. Padala, X. Z. Mustafa, et al. " Adaptive Control of Virtualized Resources in Utility Computing Environments". ACM, EuroSys'07, March 21-23, Lisboa, Portugal.
[21] M. Schmid, D. Marinescu, and R. Kroeger. " A Framework for Autonomic Performance Management of Virtual Machine-Based Services". In: Proceedings of the 15th Annual Workshop of HP Software University Assocaiton. Hosted by AI Akhawayn University in Ifran, June 22-25, 2008. Conference & Exhibition (e-2007) The Hague, The Netherlands, October, 2007.
[22] L. Grit, D, Irwin, A. Yumerefendi, J. Chase. " Virtual Machine Hosting for Networked Clusters: Building the Foundations for Autonomic Orchestration". In the Second International Workshop on Virtualization Technology in Distributed Computing (VTDC 2006).
[23] Dan Marinescu. Design and Evaluation of Self-Management Approaches for Virtual Machine-Based Environments. Master Paper, Feb. 2008.
[24] Z. Wang, X. Zhu, P. Padala, and S. Singhal. Capacity and Performance Overhead in Dynamic Resource Allocation to Virtual Containers. In the 10th IFIP/IEEE Symposium on Integrated Management, 21-25 May 2007, Munich, Germa
[25] T.Puschel, N, Borissov, M, Macias, et al. Economically Enhanced Resource Management for Internet Service Utilities. Lecture Notes on Computer Science, Vol, 4831, pp:335-348. 8th International Conference on Web Information Systems Engineering. NAcy, France, December, 2007.
[26] T. Puschel, N, Borissov, D. Neumann, et al. Extended Resource Management Using Client Classification and Economic Enhancements. 17th eChallenges e-2007
[27] M, Macias, J.O.Fito, and J. Guitart. Rule-based SLA Management for Revenue Maximisation in Cloud Computing Markets. 6th IEEE/IFTP International Conference on Network and Service Management. Niagara Falls, Canada, October, 2010.
[28] I. Goiri, F,Julia, J.I. Fito, M, Macias, and J.Guitart. Resource-level QoS Metric for CPU-based Guarantees in Cloud Providers. Lecture Notes on Computer Science (LNCS), Vol, 6296, pp:34-47. 7th International Workshop on Economics of Grids, Clouds, Systems, and Services (GECON 2010) In conjunction with the 16th European Conference on Parallel Computing (Euro-Par 2010)
[29] M,Macias, O. Rana, G. Smith, J,Guitart. Et al. Maximizing Revenue in Grid Markets using an Economically Enhanced Resource Manager. Concurrency and Computation: Practice and Experience, Vol, 22(14), pp:1990-2011. September 2010.

Xindong You is a lecturer of School of Computer Science and Technology, Hangzhou Dianzi University, China. She is with the Grid and Service Computing Lab in Hangzhou Dianzi University.

Before joining Hangzhou Dianzi University, she was a PhD candidate in Northeastern University from 2002 to 2007. She received her PhD degree in 2007. Her current research areas include virtualization, distributed computing, etc.

Dr. You is a member of China Computer Federation (CCF)

Jian Wan is the director of Grid and Service Computing Lab in Hangzhou Dianzi University and he is the dean of School of Computer Science and Technology, Hangzhou Dianzi University, China.

Prof. Wan received his PhD degree in 1996 from Zhejiang University, China. His research areas include parallel and distributed computing system, virtualization, grid computing, etc.

Prof. Wan is a member of China Computer Federation (CCF).

Xianghua Xu is an associate professor at Hangzhou Dianzi University and the vice director of Grid and Service Computing Lab in Hangzhou Dianzi University.

Dr Xu received his PhD degree in 2005 from Zhejiang University, China. His research areas include service computing, parallel and distributed computing system, virtualization, grid computing, etc.

Dr. Xu is a member of China Computer Federation (CCF) and the IEEE.

Congfeng Jiang is a lecturer of School of Computer Science and Technology, Hangzhou Dianzi University, China. He is with the Grid and Service Computing Lab in Hangzhou Dianzi University.

Before joining Hangzhou Dianzi University, he was a PhD candidate in Huazhong University of Science and Technology from 2002 to 2007. He received his PhD degree in 2007. His current research areas include power aware computing system, virtualization, grid computing, etc.

Dr. Jiang is a member of China Computer Federation (CCF), IEEE and IEEE Computer Society.

Jilin Zhang is a lecturer of School of Computer Science and Technology, Hangzhou Dianzi University, China. He is with the Grid and Service Computing Lab in Hangzhou Dianzi University.

Before joining Hangzhou Dianzi University, he was a PhD candidate in Beijing University of Technology from 2005 to 2009. He received his Ph.D. degree in 2009. His current research areas include parallel processing, complier optimization, virtualization, distributed computing, etc.

Dr. Zhang is a member of China Computer Federation (CCF) and the IEEE.