ADVANCES in NATURAL and APPLIED SCIENCES

ISSN: 1995-0772
EISSN: 1998-1090
2016 April 10(4): pages 14-21

Published BY **AENSI Publication**
http://www.aensiweb.com/ANAS
Open Access Journal

Incentive-Compatible Approximation Mechanism for Auction-Based VM Provisioning in Clouds

[1]A.Kirubavathi and [2]Dr.N.Uma Maheswari

[1]*PG Scholar, department of Computer Science and Engineering PSNA College of Engineering and Technology Dindigul*
[2]*Professor, Department of Computer Science and EngineeringPSNA College of Engineering and Technology Dindigul*

Received 25 February 2016; Accepted 10 April 2016; Available 15 April 2016

Address For Correspondence:
A.Kirubavathi, PG Scholar, department of Computer Science and Engineering PSNA College of Engineering and Technology Dindigul.
E-mail: kirubavathi93@gmail.com

ABSTRACT

Cloud providers face a high number of decision problems when offering Infrastructure as a service to their customers. One of the biggest decision problems is how to provision and allocate resources such that maximizing their profit and the resources are utilized efficiently. Cloud providers provision their resources either statically or dynamically, and then allocate them in the form of virtual machine instances to their customers. In the case of dynamic provisioning, the provider provisions the resources by considering the current users demand. Due to variable load demand, dynamic provisioning leads to high efficient resource utilization and higher revenues for the cloud provider. Recently, cloud providers introduced auction-based models, which allow users to submit bids for their requested virtual machines. In this project, dynamic virtual machine provisioning and allocation problem formulated for the auction-based model considering many types of resources by designing approximation mechanism. Proposed mechanism is incentive-compatible, that is, the users do not have incentives to change the system by lying about their requested bundles of virtual machine instances and their valuations.

KEYWORDS: Approximation mechanism; Dynamic provisioning; Incentive-Compatible; Infrastructure as a Service;

INTRODUCTION

Cloud providers form dynamically scalable resources and these resources allocated to users based on a pay-as-you-go model. These resources are provided as three types of services: Infrastructure as a Service, Platform as a Service, and Software as a Service.

Infrastructure as a Service, the capability provided to the consumer is to provision processor, storage, memory, networks and other fundamental computing resources where the consumer is able to install and run random software, which can incorporate OS (Operating System) and applications. The consumer does not manage or control the cloud infrastructure but has power over OS, storage, and deployed applications. Platform as a Service, the facility provided to the consumer is to manage onto the cloud infrastructure consumer created applications created using programming languages, services, libraries and tools supported by the provider. The consumer does not manage or control the cloud infrastructure including network, servers, OS, or storage but has power over the deployed applications and probably configuration settings for the applications-hosting environment. Software as a Service, the facility provided to the consumer is to use the provider's applications. The applications are reachable from various client devices through either a thin client interface, such as web browser. The user does not deal with or control the cloud infrastructure including network, servers, OS, storage

To Cite This Article: A.Kirubavathi and Dr.N.Uma Maheswari., Incentive-Compatible Approximation Mechanism for Auction-Based VM Provisioning in Clouds, 2016. **Advances in Natural and Applied Sciences**. 10(4); Pages: 14-21

and individual application capabilities, with the possible exception of restricted user specific application configuration settings [1].

In this paper, Infrastructure as a service is focused where the cloud providers offer multiple types of resources in the form of VM (Virtual Machine) instances. Infrastructure as a Service providers such as Microsoft Azure and Amazon EC2 (Amazon Elastic Compute Cloud) offer four types of VM instances: small, medium, large and extra large.

Cloud providers provision the resources either statically or dynamically, and then allocate them in the form of VM instances to customers. In the static provisioning, the cloud provider pre-provisions a set of VM instances without taking into account the current demand from the users, whereas in the case of dynamic provisioning, the cloud provider provisions the resources by considering present users demand. Due to the variable load demand, dynamic provisioning leads to a high efficient resource utilization and ultimately to higher revenues for the providers. The aim of this paper is to assist dynamic provisioning of multiple types of resources based on the users request.

Cloud providers can use fixed-price and auction-based models to sell the virtual machine instances to users. In the fixed-price model, the price of every type of VM instance is fixed and pre-determined by the provider, while in the case of auction-based model, every user bids for a subset of available VM instances and auction mechanism decides the price and the allocation [3]. In this paper, the design of mechanisms for auction-based model is considered. In the auction-based models, users can obtain their requested resources at lesser prices than in the case of the fixed-price models. Moreover, the cloud providers can raise their profit by allowing users to bid on unutilized capacity.

The users are also selfish in the sense that they want to maximize their own utilities by declaring false requests. It may be beneficial for them to manipulate the system by declaring dissimilar bundles or bids from their actual requests. In an untruthful auction, users may bid much lower than their actual valuations which not only may possibly hurt other users, but also may indirectly lead to profit losses for the cloud provider. Thus, unless incentive-compatible is enforced, maximizing the revenue may not be effective. In incentive-compatible auctions, the overlook strategy for users is to bid truthfully. When users report their true valuations, the cloud provider can allocate their resources efficiently to users who value them the most.

In this paper, incentive-compatible approximation mechanism is designed and that mechanism solves the VM provisioning and allocation problem. Allocation algorithm used to find the winning users. Payments charged for winners using payment function. Winning users are always considered as winners for next biddings also if they bid higher or request smaller bundle compared to previous biddings. The goal is to find an allocation of resources to the users such that maximizing the social welfare, where the social welfare is the summation of users valuations. The proposed mechanism is incentive-compatible, that is, the users do not have incentives to lie down about their requested bundles of VM instances and their bids.

Related Work:

Some researchers investigated various resource allocation problems in clouds by employing game theory. [5] designed an well-organized truthful-in-expectation mechanism for resource allocation in clouds where only one type of resource was considered. [7] shown that system heterogeneity plays an important role in determining the dynamics of truthful mechanisms. This proposed mechanism take into account the heterogeneity of the systems and that of user requests when making allocation decisions. Di Valerio *et al*. [9] formulated the service provisioning problem as a Stackelberg game, and computed the equilibrium price and the allocation strategy by solving the associated optimization problem. However, both studies considered only one type of VM instances, so, the problem they solved is a one dimensional provisioning problem. Mechanism design theory has been engaged in designing truthful allocation mechanisms in several areas. In particular, there is a large amount of work in spectrum auctions, where a government or a primary license holder sells the right to employ a specific frequency band in a specific area via auction based mechanisms (e.g., [10], [11], [12], [13]). In these studies, only one type of resource means the spectrum is available for allocation. However, in this paper, several types of resources (e.g., core, memory, storage) are considered, and thus the mechanisms proposed in the above studies cannot be used in our context. Zhou *et al*. [10] proposed a truthful mechanism that assumes the existence of k uniform channels that can be spatially reused means a channel can be allocated to more than one user simultaneously. Their greedy mechanism sorts the bidders in sliding order of their bids. Wu and Vaidya [13] extended the study of Zhou *et al*. by proposing a truthful mechanism considering assemblage the users based on their spatial conflicts. Their greedy mechanism is based on the ordering of the group's bids. But, VM instances cannot be simultaneously assigned to the users and thus, their mechanism cannot be used to solve the virtual machine allocation problem. The closest work to ours in the spectrum allocation region is by Jia *et al*. [14] who proposed truthful mechanisms for secondary spectrum market. The authors considered K uniform channels covering a certain area that is partitioned into small cells. This problem considers several cells available which in a few sense correspond to several types of VMs in our study. Nevertheless, in each cell a fixed number of uniform channels are available to be sold, whereas, in this paper, each VM instance is composed of numerous

types of heterogeneous resources. In addition, the mechanism proposed by Jia *et al*. [14] incorporates a straightforward greedy metric for order the users that is based on the ratio of their bids to the number of requested channels. However, this proposed mechanisms incorporate bid density metrics that not only consider the structure of VMs means the multiple resources, but also take into consideration the scarcity of resources. In addition, do not limit the number of available VMs for each type of VM, and dynamic provisioning of VMs allowed. The design of truthful mechanisms for resource allocation in clouds has been analyzed by Zaman and Grosu [15], [16]. They projected a combinatorial auction based mechanism, CA-GREEDY (Combinatorial Auction-GREEDY) to distribute VM instances in clouds [16]. They shown that CA-GREEDY can capably allocate VM instances in clouds generate higher revenue than the currently used fixed price mechanisms. However, CA-GREEDY requires that the virtual machines are provisioned in advance, that is, it requires static provisioning. They extended their work to dynamic scenarios by a mechanism called CA-PROVISION (Combinatorial Auction-PROVISION) [15]. CA-PROVISION selects the set of virtual machine instances in a dynamic approach which reflects the market demand at the time when the mechanism is executed. But, these mechanisms do not consider numerous types of resources. Their proposed mechanisms only consider computational resources means cores, which is only one of the dimensions in this paper. In addition to this, proposed mechanism consider the scarcity of the resources while making provisioning and allocation decisions.

I. Mechanism Design Framework:

A mechanism M = (A,P) contains an allocation function $A = (A_1,...,A_n)$ and a payment rule $P = (P_1,...,P_n)$. The allocation function determines that which users receive their requested bundles, and the payment rule determines the amount that each user must pay.

In this model, there are n users, and the type of a $user_i$ is denoted by $T_i = (R_i, b_i)$. Social welfare V, which is defined as the sum of users valuations:

$$V=\sum v_i(R_i).x_i \quad (1)$$

where x_i, i=1,...,n, are decision variables defined as, x_i=1,if bundle R_i is allocated to user I and x_i=0, otherwise.

$User_i$ desires only the requested bundle of virtual machine instances, R_i , and derives value of b_i if he gets the requested bundle or any superset of it and otherwise, zero value. Thus, the valuation function for $user_i$ as follows:

$$v_i(A_i) = b_i \quad \text{if } R_i \subseteq A_i, \text{ otherwise } 0 \quad (2)$$

The goal is to design a incentive-compatible mechanism that increases the social welfare V .The goal of a user is to maximize their utility, and may manipulate the mechanism by lying about their true type to increase their utility. In this case , since the type of a user is pair of bundle and value, the user can lie about the value by coverage a higher value in the hope to increase the likelihood of obtaining their requested bundle. These manipulations by the users will direct to inefficient allocation of resources and reduce the revenue obtained by the cloud provider. So there is a need to prevent such manipulations by scheming incentive-compatible mechanisms for solving virtual machine provisioning and allocation problem. Users have incentives to expose their true types.

A. Incentive-Compatible:

A mechanism is incentive-compatible, if reporting truthfully is a dominant strategy for the users, that is, the users maximize their utilities by reporting true type independently of the other users are being reported. To obtain a incentive-compatible mechanism the allocation function must be monotone, and the payment rule P must be based on the critical value.

B. Monotonocity:

Any winning user who receives their requested bundle by declaring a type T_i is still wining if user requests a smaller bundle, and values a higher bid.

C. Critical value:

The mechanism M works as follows. It first receives the bundles and bids from each and every participating user, and then, based on the received types determines the allocation by using the allocation function and the payments using the payment rule.

II. Incentive-Compatible Approximation Mechanism:

As shown in the incentive-compatible approximation mechanism algorithm, auction mechanism collects all the user requests. Request consist of bundle of virtual machine instances and bid price (lines 1-3). Winning users will be determined by the allocation algorithm. Vector of user requests and resources capacity are inputs for the allocation algorithm (line 4). Initially the social welfare (sum of users valuation) will be zero (line 5). If sum of user requested bundle is less than or equal to mechanism resources capacity then the social welfare will be

incremented by bid price. If user i bids the virtual machine most such that maximizing the social welfare then allocation value assigned as one otherwise social welfare will be decremented by the bid price and the allocation value will be assigned as zero. If sum of user requested bundle greater than resources capacity then that particular user request cannot be considered. Allocation algorithm returns the value of social welfare and vector of allocation (lines 6-16). If allocation value for the user is one then he will be considered as winning users. Virtual machine will be provisioned and allocated, and payments will be charged to that user. Otherwise that user cannot be considered as winning users and virtual machine provisioning will not be done to that user (lines 18-24). Incentive provided to winning users if he requests less virtual machine bundle or bids higher than previous auctions then winning users always considered as winning users otherwise they will lose the auction (lines 25-31).

Algorithm : Incentive-Compatible Approximation Mechanism

1: {Collect user requests}
2: for all $user_i$
3: $Request_i = (bundle_i, bid_i)$
4: Allocation algorithm(Vector of requests, Resources capacity)
5: Social welfare = 0
6: for all $user_i$
7: if sum of user requested $bundle_i$ <= Resources capacity
8: Social welfare=Social welfare + bid_i
9: if $user_i$ bids the VM most such that maximizing the social welfare
10: allocation = 1
11: else
12: Social welfare=Social welfare - bid_i
13: allocation = 0
14: else
15: user $request_i$ cannot be considered
16: return(Social welfare, Vector of allocation)
17: Payment function(Vector of allocation, bid)
18: for all $user_i$
19: if(allocation = 1)
20: $User_i$ considered as winning user
21: Provision and allocate VM to $user_i$
22: Payment function charges the payment for $user_i$
23: else
24: VM cannot be provisioned and allocated to $user_i$
25: for all winning $user_i$
26: Winning $user_i$ always considered as winning user for next biddings
27: if Winning $user_i$ requests less VM bundle or bids higher than previous ones
28: Provision and allocate VM to winning $user_i$
29: Payment function charges the payment for winning $user_i$
30: else
31: Winning $user_i$ cannot be considered as winning user for next bidding.

III. Model:

As shown in the Fig. 1 cloud provider starts the auction mechanism. Provider will list the available virtual machine instances i.e., core, memory, storage. Cloud users will send the request by viewing the auction mechanism generated by the cloud provider. Request consists required bundle of virtual machine instances and bid price. Different users may bid for same type of bundle so winning possibility will be low. Auction mechanism collects all the request from the users.

Auction mechanism determines the winning users by allocation algorithm. Payment charged for winning users by payment function. Both allocation algorithm and payment function are the concept of approximation virtual machine provisioning and allocation mechanism. Virtual machine will be provisioned for winning users.

Implementation:

As shown in the Fig. 2 provider and cloud users sign up for registration. Those registered accounts will be stored as database. If new user then they sign up for registration otherwise they may login to their account and use the auction system. Cloud provider creates the auction mechanism and they update the available resources capacity.

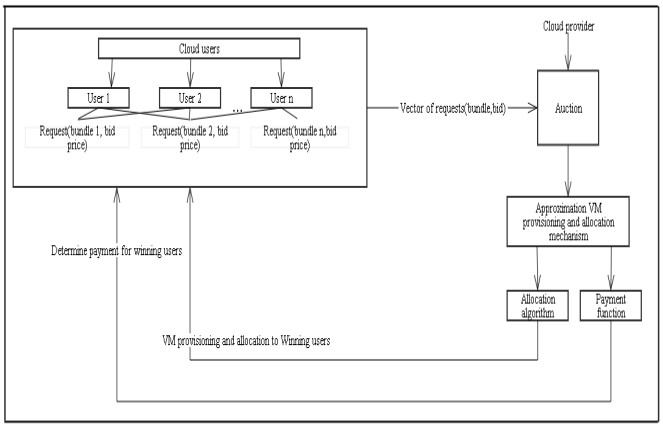

Fig. 1: Architecture of incentive-compatible approximation mechanism.

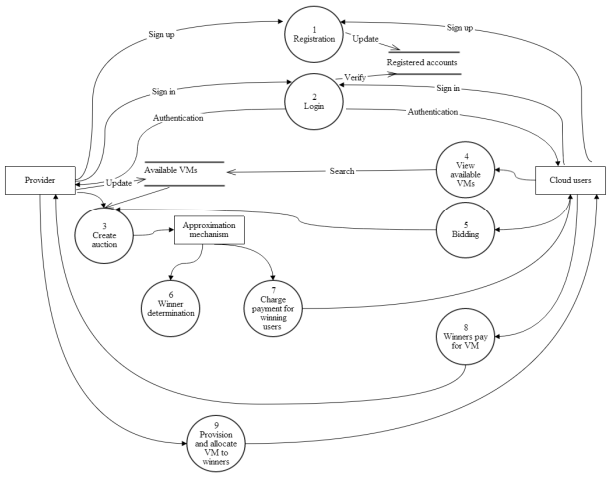

Fig. 2: DFD of incentive-compatible approximation mechanism.

Cloud users view the available resources and bid for it. Auction mechanism determines the winning users and payment for winning users using approximation mechanism. Winners pay for the virtual machine. Provider provision and allocate the virtual machine to winning users.

RESULT AND DISCUSSION

Table I shows the four types of VM instances. CPU represented as type 1 resource, memory represented as type 2 resource and storage represented as type 3 resource.

Table 1: VM Instance Types.

Types of resources	*Small*	*Medium*	*Large*	*Extra Large*
CPU	1	2	4	8
Memory(GB)	1.7	3.75	7.5	15
Storage(GB)	1	4	32	80

Our goal is to show that our proposed mechanism is robust against manipulation by a user. The true request of the eight users are shown in Table II. The capacities of the two resources are considered as follows: 100 cores and 4800 MB of storage. The incentive-compatible approximation mechanism allocates resources to user 1,2,3 and 8 in the case where all users state their true requests. The payments of the winning users calculated based on the payment function are 13,13,28 and 20 respectively.

As shown in Table III, we analyze different scenarios, where user 8 submits different requests. In addition, Fig. 3 depicts the payment and utility of the user for all the cases. In Case I, user 8 submits his true request. In this case, user 8 wins and receives the requested bundle of virtual machines, r_8. The mechanism charges her $20 for the bundle, and his utility is 90-20=70. In case II, user 8 submits a request with a higher bid B_8 =100. In this

case, user 8 is still a winning user and the mechanism determines the same payment for him as in case I, leading to a utility of 70. In case III, user submits a request with a lower bid B_8 =70, which is not less than the price determined by mechanism. Thus, user 8 is still winning user and the mechanism charges him the same amount as in case I. However, if user 8 submits a request with a lower bid below the payment, he will not obtain his requested bundle, leading to zero utility. This is shown in case IV, where user 8 submits a bid B_8 =19. We now investigate scenarios in which user 8 requests a different bundle than his true bundle. In case V, user submits a larger bundle R_8 =<4,3,2,4>, where he requests 4 VM instances of extra large type instead of 2. In this case, he obtains the bundle due to available capacities. However, he pays more than he pays in case I,II and III. Thus, his utility decreases.

Table 2: Users True Requests.

Type/User	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***
Small	0	2	3	0	3	2	3	4
Medium	0	3	0	0	0	0	0	3
Large	3	0	3	3	0	0	0	2
Extra Large	0	0	4	1	1	4	1	2
Bid	40	28	105	20	15	25	17	90

Table 3: Different Scenarios for User 8's Request Declaration.

Case	*Requested bundle(R_8)*	*Bid(B_8)*	*Scenario*	*Status*
I	<4,3,2,2>	$90	$B_{8=}\ b_8$, $R_8 = r_8$	W
II	<4,3,2,2>	$100	$B_{8>}\ b_8$, $R_8 = r_8$	W
III	<4,3,2,2>	$80	$B_{8<}\ b_8$, $R_8 = r_8$	W
IV	<4,3,2,2>	$19	$B_{8<}\ b_8$, $R_8 = r_8$	L
V	<4,3,2,4>	$90	$B_{8=}\ b_8$, $R_8 > r_8$	W
VI	<4,3,2,6>	$90	$B_{8=}\ b_8$, $R_8 > r_8$	L

In case VI, he submits a larger bundle R8 =<4,3,2,6>, where he requests 6 VM instances of type extra large instead of 2 VM instance. However, he becomes a loser since the cloud provider does not have sufficient resources to fulfill his request. Finally proved that if a user submits a request untruthfully, he can not increase his utility.

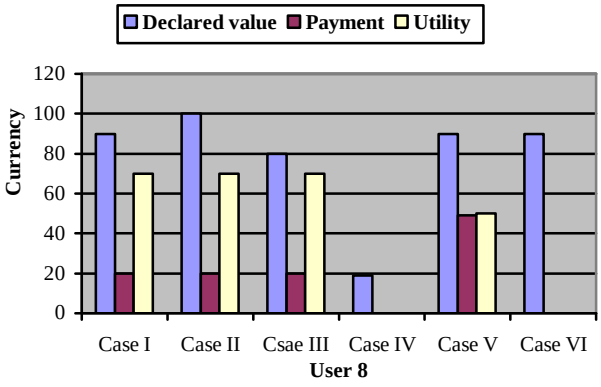

Fig. 3: Effect of untruthful declarations.

In the absence of publicly available user requests data from cloud providers, we resort to the well studied and standardized workloads as of both the Grid Workloads Archive and the Parallel Workloads Archive. We selected three logs as of the Grid Workloads Archive as follows: 1) NorduGrid traces from the NorduGrid system; 2) AuverGrid traces from the AuverGrid system; 3) SHARCNET traces from the SHARCNET clusters installed at several academic institutions in Ontario, Canada. Also, chosen the following log from the Parallel Workloads Archive, 4) MetaCentrum from the national grid of the Czech republic.

Fig. 4 shows the average social welfare per hour for the logs. In Fig. 4, GWA-T-3 NorduGrid log represented as GWA-1, GWA-T-4 AuverGrid log represented as GWA-2, METACENTRUM-2009-2 log represented as PWA, GWA-T-10 SHARCENT log represented as GWA-3 and GWA-T-10 SHARCENT (2) log represented as GWA-4. Incentive-Compatible Approximation Mechanism obtained optimal results similar to the one obtained by the optimal VCG (Vickrey Clarke Groves) for all logs. Note that Incentive-Compatible Approximation Mechanism guarantees worst case performance, and it does not essentially produce non-optimal solutions. The rounding procedure of Incentive-Compatible Approximation Mechanism makes the size of the requests larger than their actual size. For most of the cases, the total size of the requests in the optimal solutions is not equal to the available capacities. That means, there are extra remaining capacities even in the optimal allocation. As a result, the rounding of the optimal solution still fits in the existing capacity. Note that such cases

occur irrespective of the amount of sum of users demands, which could be much higher than the available capacities.

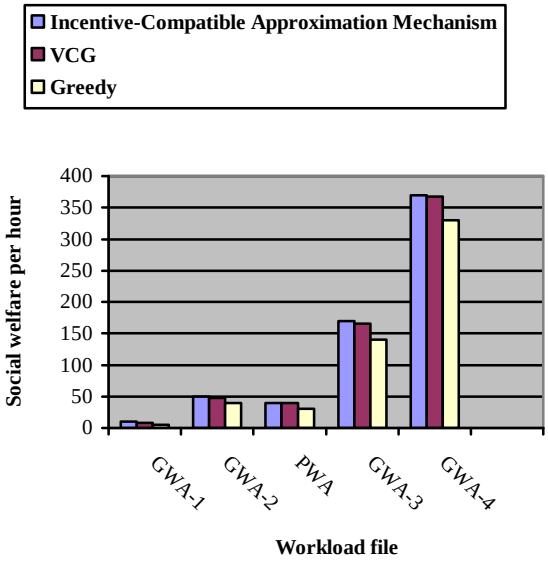

Fig. 4: Incentive-Compatible Approximation Mechanism versus VCG & Greedy : Social welfare.

The optimality of Incentive-Compatible Approximation Mechanism depends on the total size of the requests in the optimal solution and the available capacities. Incentive-Compatible Approximation Mechanism can find the optimal solution in such cases. In fact, there should be a specific configuration in the requested bundles and the available capacities in order to create the worst case scenario for the Incentive-Compatible Approximation Mechanism. Since Greedy considers only the bid densities when building allocation decisions, it obtains the lowest social welfare in general, which is far from the optimal social welfare. For example, for GWA-4, that is, GWA-T- 10 SHARCNET (2) the optimal social welfare is 362, while Greedy obtains a social welfare of 317 leading to a 12 percent gap from the optimal solution obtained by Incentive-Compatible Approximation Mechanism.

Fig. 5 shows the percentage of users that have been allocated by the mechanisms. Since the Incentive-Compatible Approximation Mechanism achieves the optimal solutions for all logs, it serves the same percentage of users as VCG. Note that VCG does not serve a higher number of users than Greedy. This is due to the truth that the optimal mechanism finds the most important subset of users in order to maximize the social welfare.

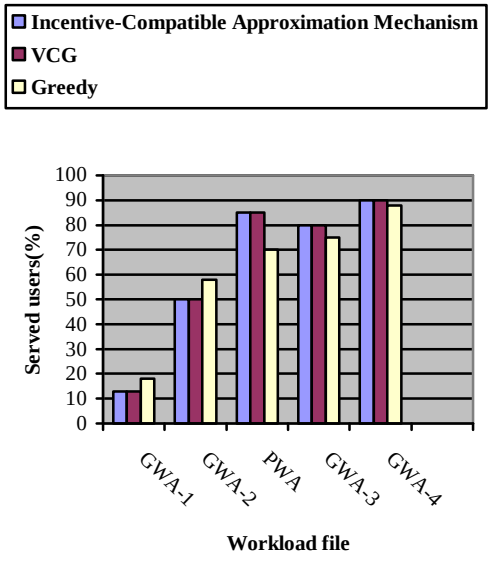

Fig. 5: Incentive-Compatible Approximation Mechanism versus VCG & Greedy: Users served.

Conclusion:

The problem of auction-based virtual machine provisioning under the concept of dynamic virtual machine provisioning, allocation and payment determination in clouds addressed by designing incentive-compatible approximation mechanisms that give incentives to the users to expose their true valuations for their requested bundle of virtual machine instances. The proposed incentive-compatible approximation mechanism for solving the virtual machine provisioning and allocation in clouds problem consider the presence of resources of multiple types. The objective of the proposed mechanism is to maximize the social welfare in dynamic resource provisioning. The result of the incentive-compatible approximation mechanism will be near optimal allocations while giving incentives to the users to report their true valuations for the bundles of virtual machine instances.

REFERENCES

1. NIST Definition of Cloud Computing, 2011. http://www.nist.gov/cloud/NIST_SP-500-291_Version-2_2013_June18..
2. Amazon EC2 Instance Types, 2014. http://aws.amazon.com/ec2/ instance-types/.
3. Nejad, M.M., L. Mashyekhy, D. Grous,2015. "Truthful Greedy Mechanisms for Dynamic Virtual Machine Provisioning and Allocation in Clouds", IEEE Transactions on parallel and distributed systems.
4. Wei, G., A. Vasilakos, Y. Zheng and N. Xiong, 2010. "A Game-Theoretic Method of Fair Resource Allocation for Cloud Computing Services," The J. Supercomputing, 54(2): 252-269.
5. Jain, N., I. Menache, J. Naor and J. Yaniv, 2013. "A Truthful Mechanism for Value-Based Scheduling in Cloud Computing," Theory of Computing Systems, pp: 1-19, http://dx.doi.org/10.1007/s00224-013- 9449-0.
6. Kong, Z., C.Z. Xu and M. Guo, 2011. "Mechanism Design for Stochastic Virtual Resource Allocation in Non-Cooperative Cloud Systems," Proc. IEEE Fourth Int'l Conf. Cloud Computing, pp: 614-621.
7. Wang, Y., A. Nakao and A.V. Vasilakos, 2012. "Heterogeneity Playing Key Role: Modeling and Analyzing the Dynamics of Incentive Mechanisms in Autonomous Networks," ACM Trans. Autonomous and Adaptive Systems, 7(3): 31.
8. Ardagna, D., B. Panicucci and M. Passacantando, 2013. "Generalized Nash Equilibria for the Service Provisioning Problem in Cloud Systems," IEEE Trans. Services Computing, 6(4): 429- 442.
9. DI valerio, V. cardellini, and F. LO Presti, 2013. "Optimal Pricing and Service Provisioning Strategies in Cloud Systems: A Stackelberg Game Approach," Proc. IEEE Sixth Int'l Conf. Cloud Computing, pp: 115-122.
10. Zhou, X., S. Gandhi, S. Suri and H. Zheng, 2008. "ebay in the Sky: Strategy-Proof Wireless Spectrum Auctions," Proc. ACM MobiCom, pp: 2-13.
11. Zhou, X. and H. Zheng, 2009. "Trust: A General Framework for Truthful Double Spectrum Auctions," Proc. IEEE INFOCOM, pp: 999-1007.
12. Kasbekar, G.S., S. Sarkar, 2010. "Spectrum Auction Framework for Access Allocation in Cognitive Radio Networks," IEEE/ACM Trans. Networking, 18(6): 1841-1854.
13. Wu, F., N. Vaidya, 2013. "A Strategy-Proof Radio Spectrum Auction Mechanism in Noncooperative Wireless Networks," IEEE Trans. Mobile Computing, 12(5): 885-894.
14. Jia, J., Q. Zhang, Q. Zhang and M. Liu, 2009. "Revenue Generation for Truthful Spectrum Auction in Dynamic Spectrum Access," Proc. ACM MobiHoc, pp: 3-12.
15. Zaman, S. and D. Grosu, 2011. "Combinatorial Auction-Based Dynamic VM Provisioning and Allocation in Clouds," Proc. IEEE Third Int'l Conf. Cloud Computing Technology and Science, pp: 107-114.
16. Zaman, S., D. Grosu, 2013. "Combinatorial Auction-Based Allocation of Virtual Machine Instances in Clouds," J. Parallel and Distributed Computing, 73(4): 495-508.
17. Kellerer, H., U. Pferschy and D. Pisinger, 2004. Knapsack Problems. Springer.
18. Mu'alem and N. Nisan, 2002. "Truthful Approximation Mechanisms for Restricted Combinatorial Auctions," Proc. 18th Nat'l Conf. Arti- ficial Intelligence, pp: 379-384.
19. Lehmann, D., L. Ocallaghan and Y. Shoham, 2002. "Truth Revelation in Approximately Efficient Combinatorial Auctions," J. ACM, 49(5): 577-602.
20. Nejad, M.M., L. Mashayekhy and D. Grosu, 2013. "A Family of Truthful Greedy Mechanisms for Dynamic Virtual Machine Provisioning and Allocation in Clouds," Proc. IEEE Sixth Int'l Conf. Cloud Computing, pp: 188-195.