2016 International Conference on Circuit, Power and Computing Technologies [ICCPCT]

Auction Based Dynamic Resource Allocation in Cloud

E. Iniya Nehru
Senior Technical Director,
National Informatics Center,
Chennai, India
Email: nehru@nic.in

Infant Smile Shyni. J
PG student,
TIFAC – CORE in PCT,
Velammal Engineering College,
Chennai, India
Email: infantsherline19@gmail.com

RanjithBalakrishnan
Assistant Professor,
TIFAC – CORE in PCT,
Velammal Engineering College,
Chennai, India
Email: ranjithbke@gmail.com

***Abstract*—Cloud computing is one of the fastest emerging technology. As we all know, cloud is a wide pool of resource which provides resources based on the users request. Any service can be provided as a service through cloud. As the cloud contains many resources, there may also wastage of resources. To reduce this wastage, cloud providers enter into auctioning of resources when the demand is high. Cloud computing includes distinct resources. Because of the complementary and supplementary effects between distinct assets, bidders have preferences not for just a single resource but also for a set of resources. Auctioning for a bundle of resources is called combinatorial auction. Dynamic resource allocation in on demand for a bundle of resource is proposed by using CA – PROVISION algorithm and the scenarios are simulated using Cloudsim, a simulator meant for cloud computing analysis.**

Keywords—Cloud Computing, Auction, Bid, Dynamic Resource Allocation, Combinatorial auction and Cloudsim.

I. INTRODUCTION

Cloud computing is the fastest emerging paradigm of distributed computing. It is a computing model and not technology. It is an internet based computing where users have to pay as per their usage. Cloud computing has many advantages like no infrastructure investment, storage capability, scalability, backup and recovery capabilities, anytime access with internet, cost effective, etc., Cloud provides many types of services that comes under the major categories like infrastructure, software and platform. As cloud computing reduces the installation costs, many of the users are switching over to cloud computing resources. Now a day's anything i.e. data, storage, monitoring, network etc., can be provided as a service via cloud computing which is denoted as XaaS (Anything-as-a-Service) [3].

Cloud computing is a virtualized environment which provides services as Virtual Machine (VM) instances. These VM instances are distinct. Cloud users can choose which configuration or amount of instances they want. As there are many resources available, there may be some situations which lead to underutilization of these resources. So, a proper resource allocation technique has to be done to increase the revenue of the cloud provider.

II. PROBLEMS IN RESOURCE ALLOCATION

Cloud computing is mainly popular for flexibility which can be attained through scalability and availability of services whenever in need. It aims to supply reliable, customized and good QoS parameters in the dynamically changing computing environments. To attain these goals, resource allocation is very much important. Proper resource allocation gains customer satisfaction also.

A. Significance in resource allocation

Resource allocation in cloud is the process of assigning cloud resources to the requested cloud users over the internet. Resource allocation problems are [13]:

a) Resource contension: More users tries to access the same resource thereby increasing traffic.
b) Resource scarcity: When there are no enough resources.
c) Resource fragmentation: It leads to isolation of resources to some users and they can't be allocated to the persons in need.
d) Over provisioning: Providing more resources than the need.
e) Under provisioning: Providing less resources than the requested amount.

To overcome these issues of resource allocation in cloud computing, auction mechanism is used.

B. Resource allocation by auction

Auction is a process of buying and selling goods by making bids for the available resources. The highest bidder is provided with the resource. Auction comes into play when the demand is high and resources are less. Both the buyer and the seller are free to provide their valuation for the resource they are in need and by iterative discussions, an agreement is made for the payment of the resource. As in cloud computing, the wide pools of resource are not used by the users to its full utility. Also in on demand situation, under utilization of resources can happen if we follow the FCFS (First Come First Serve) basis. So, if we use auction mechanism for resource allocation, utilization of resources as well as providers has economic benefits and proper allocation reduces wastage of resources [12].

Auction environment includes auctioneer, broker, cloud provider and user. Auctioneer may be the broker sometimes. This is depicted in the Fig. 1 below [1].

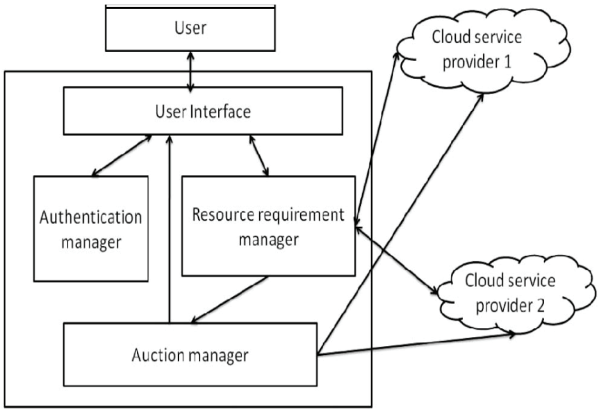


Figure 1: Simple auction environment in cloud

Many existing techniques in auction use static allocation and pricing. The problems in static allocation are:

i. Contracts in resource allocation always favor the vendors. There may be some situations when the instances of users and vendors are mismatched.

ii. It is very difficult to maintain the SLA in fixed pricing.

These problems can be overcome by dynamic resource allocation. Dynamic resource allocation provides economic benefit to the provider and the client has to pay only for the usage of the resources. Truthfulness can be attained. In auction winner determination is the major issue as it is NP – hard problem.

Through virtualization, cloud provider can be able configure the VMs based on the user's preferences. Any combinations of VM's can also be obtained with the available resources with the provider. Instead of single resource, auction for this bundle of resources or combinations is called combinatorial auctions [7].

Organization of this paper is given by:In Section II, problems in resource allocation is discussed. Section III deals with related works. Section IV explains the combinatorial auction algorithm and Section V provides the simulation results. In Section VI future enhancements of this work are discussed.

III. RELATED WORKS

VM provisioning by auction is experimented by many researchers. Dynamic provisioning of resources was done by Quiroz et al. through decentralized online clustering algorithm in the grid environments. This allows users to choose their own combinations of resources. Dynamically changing environment is the base problem. Our proposed system deals with the current demands from the users [4]. Combinatorial auction based protocol for resource allocation in grids is well explained by Anubhav Das and Daniel Grosu[5].Their work explains about the winner determination problem through approximation algorithms for auctions. This involves the sequence of greedy procedures for allocation of resources which favors only to the cloud provider's profit. Daniel Lehmann et al.[6] proposed greedy method for winner determination for single minded auctions. Bidders are prioritized based on the norms in descending order and no two bidders have the same norm. Deadlock condition is also checked after each allocation.

Truthful character of the algorithm is explained by AhuvaMualem et al.[2]This provides the idea that, whenever the value of the resource increases, the usage cost for that resource also increases. This deals with the dynamic pricing mechanism of resources. SharrukhZaman and Daniel Grosu[9] proposed the CA – LP and CA – GREEDY mechanisms of combinatorial auction to find out the winner and for dynamic allocation. This algorithms uses fixed pricing mechanisms as it follows static allocation of resources. CA – LP mechanism involves solving of auction by linear programming. CA – GREEDY algorithm uses is the extended version of greedy allocation as it favors only the providers. Recent works of them[10] reveals the CA – PROVISION algorithm which is the density based dynamic resource allocation. Our proposed work eliminates the dummy user concept and reverses price methods in CA – PROVISION algorithm. Implementation is done by using Cloudsim simulator.

IV. COMBINATORIAL AUCTION BASED DYNAMIC RESOURCE ALLOCATION

Combinatorial auction allows the users to bid for a bundle of resources or packages. Bidding amount is concerned for the whole package.

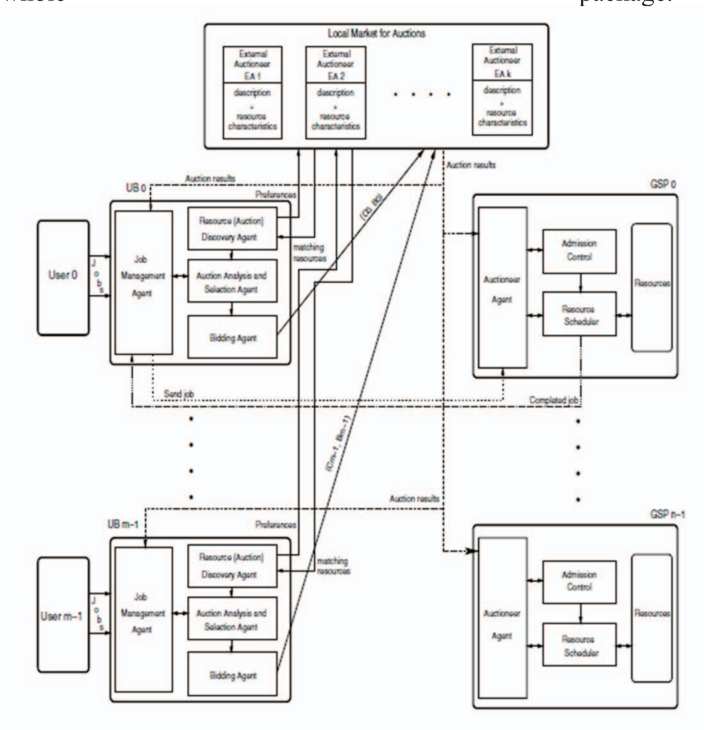


Figure 2: Combinatorial auction

In some cases, cloud provider itself provides some default packages like in Amazon [3] and users can also have the preferences for the packages of their combinations.

A sample combinatorial auction mechanism is depicted in Fig. 2 [5]. Allowing the users to express their preferences increases the provider's repudiation and revenue. Combinatorial auction mechanism is mainly meant for improving economic efficiency and system performance [11].

In this work, when the resource contention or the competition for the resource increases, the cloud manager will call for the auction of resources among the users. Users receive the notification and then start to send bids for the resources. This combinatorial auction based mechanism called CA – PROVISION provides the information about the allocation of available resources to the requested users, amount to be paid by the users, set of VM instances allocated to each user and this mechanism mainly follows the all or none format. Either the set of requested resources are allocated or none of them is allocated to the users. Users can request for their own combinations of VMs through the bids.

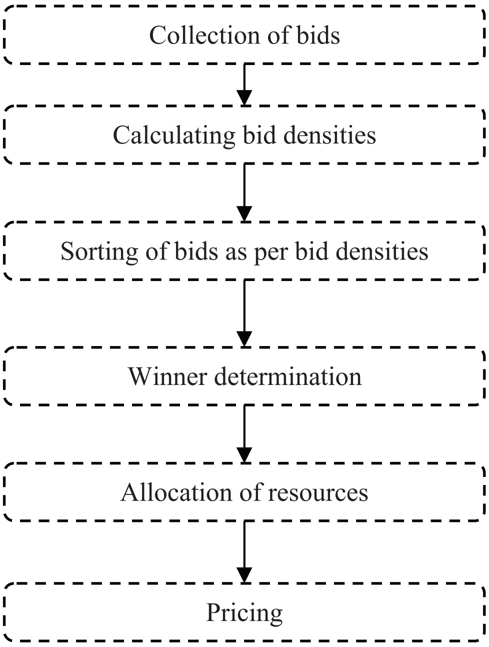


Figure 3: Flow chart for CA – PROVISION algorithm

This algorithm is divided into three major phases: Collecting the bids from the users, winner determination by density calculation and payment. A cloud provider might consist of *n* different types of VM instances and its weight based on computing power is indicated by *w*. Cloud providers decides the weights for the resources during the configuration of the resources itself. Users can able to choose any type of VM instances they want and bid for the resource in combinations as

$$B_j = (r_1^j, \dots, r_m^j, v_j)$$

This indicates the user *j* is requesting the bid containing the *m* different types of resources they are in need. In this bid request, v_j is the amount the user wishes to pay for the requested resource.

Winner determination is done based on the calculation of densities of bids. Density of the bid is calculated by using the formula,

$$d_j = v_j / s_j$$

where v_j represents the amount the user wishes to pay for the resource and s_j is given by the number of instances of resources allocated and the weights of the resources. The bid density is how much amount the user wishes to pay for single unit of resource in need.

The densities of the users are sorted in descending order. Users with the highest bid are assigned the winner and the winners are sorted in the list based on their densities. Provisioning of resources is done on the basis of this sorted order in the all or none basis. After allocation of resources for a user the environment is checked for any deadlock condition. If it is found then the rollback operation is implemented. Once all the resources are provided as VM instances to the users, the payment calculation phase starts based on the usage of the resources. The users are constantly monitored and once their consumption is over they are directed to the payment gateway. Cost for owned resources which are idle is also considered. Cost calculation for both idle and consumed resources is calculated by

$$\Pi = \sum_{j=1}^{n} x_j p_j - c_R \sum_{j=1}^{n} x_j s_j - c_I \left(M - \sum_{j=1}^{n} x_j s_j \right)$$

Where C_R and C_I represents the cost of resource at run time and idle time respectively.

As this algorithm follows dynamic allocation, same resource can be allocated to the other user when it is not in use by the owned user. This increases the provider's revenue and repudiation.This CA – PROVISION algorithm is truthful as it calculates the amount only based on the users usage only.

V. EXPERIMENTAL RESULTS

CloudSim simulator is used for the simulation of cloud environment. It supports modeling and simulation of large scale cloud computing data centers and supports simulation of virtualized hosts [8].

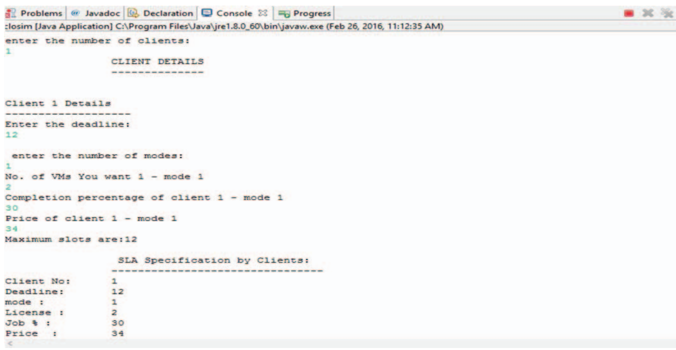
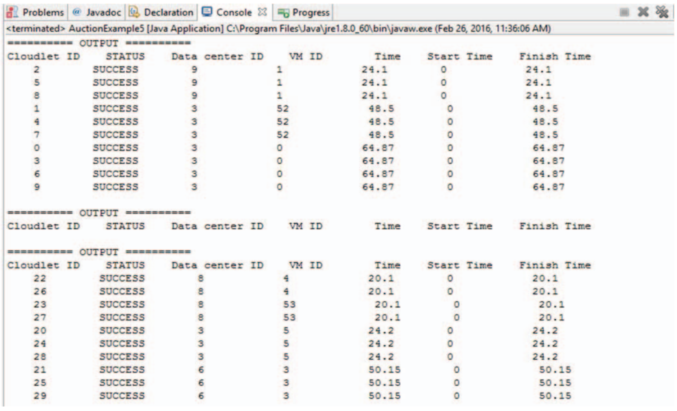


Figure 4: Density Calculation

The major usage of Cloudsim in this project is Cloudsim contains packages for implementing auctions. In this simulation, five bidders and brokers are taken and the algorithm is applied and results are studied. Results shown that the provider's revenue is increased to certain considerable extend than the normal auctions.



```
Problems  Javadoc  Declaration  Console  Progress
<terminated> AuctionExample5 [Java Application] C:\Program Files\Java\jre1.8.0_60\bin\javaw.exe (Feb 26, 2016, 11:36:06 AM)
========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID       Time    Start Time    Finish Time
    2         SUCCESS         9              1          24.1        0            24.1
    5         SUCCESS         9              1          24.1        0            24.1
    8         SUCCESS         9              1          24.1        0            24.1
    1         SUCCESS         3              52          48.5       0             48.5
    4         SUCCESS         3              52          48.5       0             48.5
    7         SUCCESS         3              52          48.5       0             48.5
    0         SUCCESS         3              0          64.87       0             64.87
    3         SUCCESS         3              0          64.87       0             64.87
    6         SUCCESS         3              0          64.87       0             64.87
    9         SUCCESS         3              0          64.87       0             64.87

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID       Time    Start Time    Finish Time

========== OUTPUT ==========
Cloudlet ID    STATUS    Data center ID    VM ID       Time    Start Time    Finish Time
    22         SUCCESS        8              4          20.1        0            20.1
    26         SUCCESS        8              4          20.1        0            20.1
    23         SUCCESS        8              53          20.1        0             20.1
    27         SUCCESS        8              53          20.1        0             20.1
    20         SUCCESS        3              5          24.2        0            24.2
    24         SUCCESS        3              5          24.2        0            24.2
    28         SUCCESS        3              5          24.2        0            24.2
    21         SUCCESS        6              3          50.15        0             50.15
    25         SUCCESS        6              3          50.15        0             50.15
    29         SUCCESS        6              3          50.15        0             50.15
```

Figure 5: Output for the simulation

Client's request traffic is also reduced due to the combinations of resource packages rather than single resource request. Dynamic resource allocation increases the cloud users and provider's repudiation. Underutilization of resources is minimized. Further implementations on increasing the number of bidders and datacenters are under progress.

VI. CONCLUSION AND FUTURE WORK

Results indicated that this CA – PROVISION algorithm can generate better revenue than the other algorithms as it is based on dynamic allocation of resources. Provider's repudiation also increased due to the truthfulness of the algorithm. Further processes are under execution. Efficient pricing mechanisms need to be studied in the future as there may occur some fraudulent cases as the cloud environment is mainly based on the internet. Auction along with VM migration techniques are to be studied further. In this paper resource allocation by auction is mainly concentrated. But when implementing in real time, the major cloud computing issues like security, privacy and heterogeneity should be addressed without affecting the performance and availability.

REFERENCES

[1] Abinandan S. Prasad and Shrisha Rao, "A Mechanism Design Approach to Resource Procurement in Cloud Computing", Proc. IEEE Transactions on Computers, pp. 17-30, 2014.

[2] Ahuva Mualem and Noam Nisan,"Truthful approximation mechanisms for restricted combinatorial auctions,"Proc. 18th Nat'l conf. Artificial Intelligence, pp. 379-384,2002.

[3] Amazon, "Amazon Elastic Compute Cloud (Amazon EC2)," http://aws.amazon.com/ec2/,2013.

[4] A. Quiroz, H. Kim, M. Parashar, N. Gnanasambandam, and N. Sharma, "Towards Autonomic Workload Provisioing for Enterprise Grids and Clouds," Proc. IEEE/ACM 10th Int'l Conf. Grid Computing, pp. 50-57, 2009.

[5] A. Das and D. Grosu, "Combinatorial Auction-Based Protocols for Resource Allocation in Grids," Proc. 19th Int'l Parallel and Distributed Processing Symp., Sixth Workshop Parallel and Distributed Scientific and Eng. Computing, 2005.

[6] D. Lehmann, L. I. OÇallaghan, and Y. Shoham, "Truth Revelation in Approximately Efficient Combinatorial Auctions," J. the ACM, vol. 49, no. 5, pp. 577-602, 2002.

[7] P. Cramton, Y. Shoham, and R. Steinberg, Combinatorial Auctions. MIT Press, 2005.

[8] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose and Rajkumar Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Proc. Wiley Online Library, 2010.

[9] S. Zaman and D. Grosu, "Combinatorial Auction-Based Allocation of Virtual Machine Instances in Clouds," Proc. IEEE Second Int'l Conf. Cloud Comp. Technology and Science, pp. 127-134, 2010.

[10] S. Zaman and D. Grosu, "A Combinatorial Auction-Based Mechanism for Dynamic VM Provisioning and Allocation in Clouds," Proc. IEEE Transactions on Cloud Comp., pp. 129-141, 2013.

[11] Sven de Vries and Rakesh Vohra, "Combinatorial Auctions: A survey".

[12] Youwen Lan, Weiqin Tong, Zongheng Liu, Yan Hou, "Multi-Unit Continuous Double Auction Based Resource Allocation Method," Int'l Conf. Intelligent Control and Information Processing, pp. 773-777, 2012.

[13] V. Vinothina, Dr. R. Sridaran and Dr. Padmavathi Ganapathi, "A Survey on Resource Allocation Startegies in Cloud Computing," Proc. Int'l Journal of Advanced Computer Science and Application, pp. 97-104, 2012.