Dynamic Resource Provisioning using Multi-Attribute Combinative Double Auction (MACDA) with SLA in Cloud Computing

P. Vinothiyalakshmi[1], R. Anitha[2]

[1]Assistant Professor, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Chennai, Tamilnadu, India.
[2]Professor & Head, Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Chennai, Tamilnadu, India

---***---

Abstract - *Cloud computing is a growing technology where the resources are provided as a service. There are huge amount of different types of resources available in cloud environment. So it is difficult to match the customer's request with available resources based on the expectations of customers and providers in cloud environment. This paper proposes the resource provisioning using auction based technique to fulfill the expectations of both customers and providers in an efficient way in cloud environment. In the proposed work the Multi-Attribute Combinative Double Auction (MACDA) resource allocation is used for auction to the customer's bids with the provider's bids by the cloud auctioneer for finding the best request-resource pairs in the cloud computing environment. The experimental result demonstrates that the proposed Multi-Attribute Combinative Double Auction (MACDA) resource allocation algorithm performs efficiently than the existing Combinatorial Double Auction Resource Allocation (CDARA) model.*

Key Words: cloud computing, multi-attribute, combinative, double auction, resource allocation

1. INTRODUCTION

The cloud computing is the best suitable place for readymade resource services. In the cloud computing environment, lot of heterogeneous resources is available and large amount of workloads are submitted by the customers simultaneously. So it is difficult to match the best workload resource pair with in a particular time. For this, in our proposed model, the auctioneer conduct the auction to find the best workload resource pair between cloud customers and providers based on Multi-Attribute Combinative Double Auction (MACDA) resource allocation. In this method, Combinative means the auction is for combination of heterogeneous resources and Double Auction indicates the auction is for both providers and customers that is., both providers and customer submit their bids to auctioneer for auction and Multi-attribute defines the heterogeneous resources considered with their more than one attributes. This auction is used to find the best workload resource pairs in an efficient way and finally the auctioneer calculate the final trade price for the allocated resources, which is pay by the customers to the providers for their allocated resources.

2. RELATED WORKS

In this section the review of existing works is done on the concept of cloud auction is as follows: GauravBaranwal et al., [1] implemented a multi-attribute combinatorial double auction for the allocation of Cloud re-sources, which not only considers the price but other quality of service parameters also. Sharrukh Zaman et al., [2] presented the combinatorial auction-based mechanisms for solving the problem of allocating VM instances in cloud. Auction-based mechanism for dynamic VM provisioning and allocation that takes into account the user demand, when making provisioning decisions yields higher revenue for the cloud provider and improves the utilization of cloud resources. Ruichun Tang et al., [3] proposed the Credibility-based cloud media resource allocation algorithm, which is similar to the continuous double auction mechanism, there source applicants and resource owners submitted their requests to the allocation agents. Based on the total credibility, the allocation agents allocated the media resources to get the optimal allocation sequence for higher allocation efficiency and Quality of Service(QoS). Weiwei Kong et al., [4] proposed a novel adaptive VM resource scheduling algorithm based on auction mechanism in which the auction was performed by considering multiple factors including network bandwidth and auction deadline. Seyedeh Aso et al., [5] explained a combinatorial double auction-based market in which a broker performs the allocation of the providers' VMs according to the users' requests. It used to maximize the total profit of users and providers by considering truthfulness, fairness, economic efficiency and allocation efficiency in resource allocation. The literature survey and the comparative study of auction mechanisms in cloud computing reveal that double auction is used for different problems in cloud computing. Some of the works used the double auction for resource allocation problem where as some of the works used the double auction for resource pricing. Some works have done for the resource allocation and pricing with the help of double auction mechanisms. Most of the works do not consider the complexity (heterogeneous resources, QoS provisioning etc.) in double auction. The proposed MACDA model considers the QoS attributes and heterogeneous resources in cloud environment.

3. SYSTEM ARCHITECTURE

Fig.1. shows the architecture of the proposed Multi-Attribute Combinative Double Auction (MACDA) resource allocation in cloud environment. In the proposed model, the customers submit their workloads to cloud auctioneer via Cloud Information Service (CIS). Then the cloud auctioneer perform the Multi-Attribute Combinative Double Auction (MACDA) with the cloud provider's bids to find the best request-resource pairs for resource allocation.

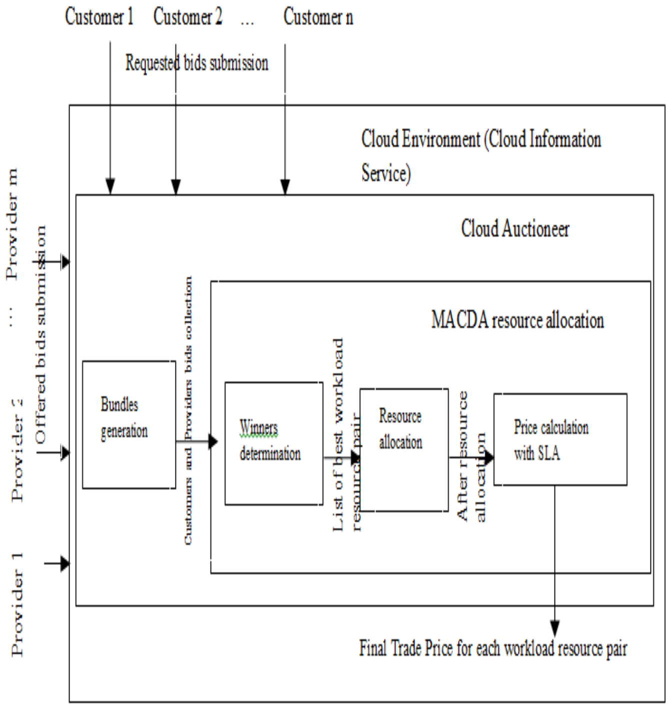

Fig. 1. Architecture of the proposed MACDA model

3.1. Multi-attribute Combinative Double Auction (MACDA) resource allocation

The customer's workloads are sending to cloud auctioneer for Multi-Attribute Combinative Double Auction (MACDA) for resource allocation. The proposed MACDA model is an extension of the existing model to combinative double auction in which there are multiple providers and customers are considered with SLA (Service Level Agreement) to fulfill the providers and customers expectations.

3.1.1. Winner determination

After receiving the bidding details for all the cloud customers and providers, the cloud auctioneer calculate the bid density value for them. Based on the bid density values, the cloud customers and providers are ranked in descending order and ascending order accordingly. This will be prioritizing the requests of the customers whose bid density vales are higher and offers of the providers whose bid density values are lower. The customer's bid density is calculated by,

$$T_c^j \leftarrow \sum_{y=1}^{l} (w_y^j * q_y^j)$$

$$Bd_c^j \leftarrow \left(\frac{p_c^j}{\sqrt{T_c^j}} \right) * t_c^j$$

And the provider's bid density is calculated by,

$$T_p^i \leftarrow \sum_{y=1}^{l} (w_y^i * q_y^i)$$

$$Bd_p^i \leftarrow \left(\frac{P_p^i}{\sqrt{T_p^i}} \right)$$

Here, the attribute '*w*' is the weight of the resource instance, which represents the capability of resource instances such as, processing power of the CPU, the storage capacity, the bandwidth or the memory capacity. Since the cloud customer's and provider's requests and offers the different resource instances with different quantities and attributes (capabilities), they should be sorted based on their requested or offered resource instances with different quantities and attributes. This leads to calculation of '*T*' which is used to convert the collection of resource instances with different capabilities into the total number of resource instances with same capability.

3.1.2. Resource allocation

After winner determination phase, the winners are found in an auction and the cloud auctioneer check the first provider from the providers list with first customer from the customers list, whether the cloud provider satisfy the customer's request or not. The customer's requested resource types are checked one after another with the provider's offer. The resource provision is the allocating resource type, its attribute and its quantity should be equal to or greater than the requested capability. If yes, the resource type will be allocated. Else the allocating process repeated for all the remaining provider's offers according to the order in the providers list, until the customer's requests are satisfied. If the customer's request is satisfied, then that customer is a winner in the current round. Otherwise, that customer is rejected in the current round. The same process will be repeated for all the remaining customers according to the order in the customers list. Finally, the cloud auctioneer will send the result of the auction to all the customers and providers who are involved in that auction for resource provision and task execution.

3.1.3. Price calculation with SLA

The average price for each customer and provider will be calculated by dividing their bid value by the total number of required or offered resource instances respectively.
The customer's average price is calculated by,

$$tl_c^j \leftarrow \sum_{y=1}^{l} (w_y^j * q_y^j)$$

$$a_c^j \leftarrow \left(\frac{P_c^j}{tl_c^j}\right)$$

The provider's average price is calculated by,

$$tl_p^i \leftarrow \sum_{y=1}^{l}(w_y^i * q_y^i)$$

$$a_p^i \leftarrow \left(\frac{P_p^i}{tl_p^i}\right)$$

The price paid by the customer to the provider for the single resource instance is calculated by,

$$pp_j^i = a_p^i * aq_j^i \quad , where \quad aq_j^i = \sum_{y=1}^{l}(w_y^j * q_y^j)$$

When the customer submits their workload to cloud providers, the provider will check the request with available resources. If the available resources satisfy the customer request then the workloads execution time is calculated. If the deadline is greater than the estimated execution time then the final price will be calculated. If the requested resources are not available or the estimated execution time exceeds deadline, then the cloud provider has to pay penalty due to SLA violation.

If (available resources < request) then
penalty = ∑ (weight of uri × quantity of uri × time duration of uri) (provider side)
else if (estimated execution time > deadline) then
penalty = ∑ (weight of ri × quantity of ri × time duration of ri) (user side)
else
penalty = 0

The final trade price paid by the customer to the provider for the collection of resource instances is calculated by,

$$FTP_j^i = \sum_{y=1}^{l} pp_y^j \pm Penalty$$

Finally the cloud auctioneer send the best workload resource pairs with final trade price details to both the respective customer and provider pairs for final payment.

4. IMPLEMENTATION

4.1. Experimentation Results

The proposed Multi-Attribute Combinative Double Auction (MACDA) resource allocation algorithm is implemented CloudAuction simulator. The performance of MACDA is compared with Combinatorial Double Auction Resource Allocation (CDARA) model which is the existing model for combinatorial double auction with multiple attributes. The MACDA is better than CDARA, because in MACDA model, the total number of resource instances for heterogeneous resources were calculated from weight of that resource instances and its quantity, in price calculation both the customers and providers are satisfied with their resources and penalty calculation used for providers and customer satisfaction in auction. In the existing CDARA model, all these factors are not considered.

Example 1:

The performance of MACDA is evaluated by preparing the scenario in which 15 participants attended an auction where 10 of them were cloud customers and 5 of them were cloud providers. In the proposed model, the customers 8, 9, 3, 6, 2, 4 were the winning customers and all of the providers were winning providers. Here the customers 10, 7, 1, 5 were rejected in the current round. The resource allocation and price payment information's are given in the tables. In the experiment, only CPU is considered for cost of resource instances on the basis of Million Instructions Per Second (MIPS). The other remaining resource instances used only for request matching purposes.

4.1.1. Winner determination and Resource allocation

Based on the bid density vales, the cloud customers and providers are ranked in descending order and ascending order accordingly. Then the cloud auctioneer do the winner determination process based on MACDA. Finally the cloud auctioneer releases the winners list (beast workload resource pair) to the participants of auction. According to the result of auction the providers provide their resources to their customers. In our example the result of MACDA winner determination is shown in the following table.

Table 1: The result of MACDA winner determination

Cloud customers (resource type, quantity)	Cloud providers (resource type, quantity)
C_8{(medium, 1), (large, 1), (very large, 1)}	P_2{(large, 1)}, P_3{(medium, 1)}, P_5{(very large, 1)}
C_9{(very large, 3)}	P_4{(very large, 2)}, P_1{(very large, 1)}
C_3{(small, 1), (very large, 2)}	P_2{(small, 1)}, P_1{(very large, 2)}
C_6{(medium, 1), (large, 1)}	P_3{(medium, 1)}, P_3{(large, 1)}
C_2{(medium, 2), (large, 1)}	P_3{(large, 1)}, P_1{(medium, 2)}
C_4{(medium, 1), (large, 1), (very large, 1)}	P_3{(large, 1)}, P_1{(medium, 1)}, P_1{(very large, 1)}

4.1.2. Price calculation with SLA for winning customer – provider pairs

In the scenario 15 participants attended an auction where 10 of them were cloud customers and 5 of them were cloud providers. In the auction, the customers 8, 9, 3, 6, 2, 4 were the wining customers and all of the providers were wining providers. Here, the customers 10, 7, 1, 5 were rejected in the current round. Since they are in the bottom of the customer's list. Still the providers P_4 and P_1 resources were not consumed completely. Because they are in the bottom position of the provider's list. After the winner determination, the auctioneer do the price calculation with

SLA for the allocated resources based on MACDA. Then the calculated amount information's send to customer and provider pairs for payment. In our example the result of MACDA price calculation values are shown in the following tables.

Table 2: The requested and paying price details of the cloud customers

Cloud Customer	Requested price	Paying price	Status
C_8	0.09	0.09	Satisfied
C_9	0.21	0.21	Satisfied
C_3	0.13	0.13	Satisfied
C_6	0.03	0.03	Satisfied
C_2	0.04	0.056	Not satisfied
C_4	0.10	0.10	Satisfied

Table 3: The offered and receiving price details of the cloud providers

Cloud Provider	Offered price	Receiving price	Status
P_2	0.03	0.03	Satisfied
P_3	0.08	0.11	Satisfied
P_5	0.06	0.05	Not satisfied
P_4	0.23	0.24(remaining resources exists)	Satisfied
P_1	0.33	0.29(remaining resources exists)	May be satisfied

The Fig. 2. shows, all the winner customers who are involved in the auction, gained the positive utility except the customer C_2. Because customer C_2 was close to the bottom of the customers sorted list and its bid value is lesser than true valuation.

The Fig. 3. shows, all the winner providers who are involved in the auction, gained the positive utility except the providers P_1 and P_5. Because they were close to the bottom of the providers sorted and their bid values are higher than true valuation. Here the providers P_1 and P_4 resources were not consumed completely.

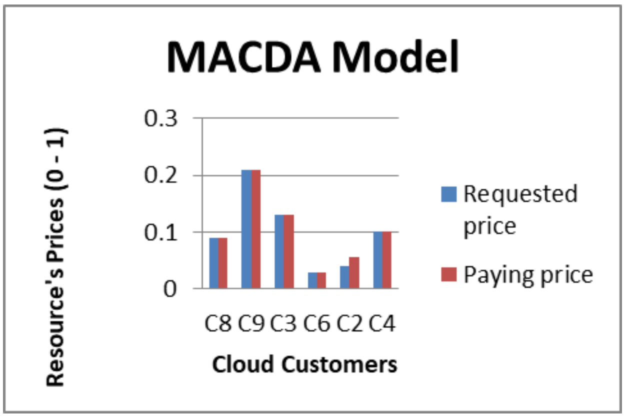

Chart -1: The proposed MACDA model for cloud customers

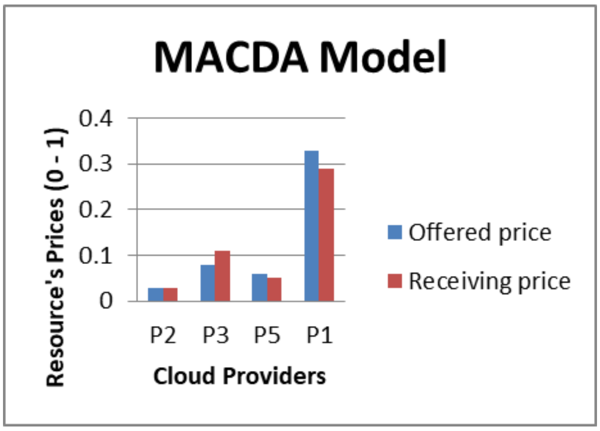

Chart -2: The proposed MACDA model for cloud providers

5. CONCLUSION

In cloud environment, there are huge amount of different types of resources available in cloud environment. So it is difficult to match the customer's request with available resources based on the expectations of customers and providers in cloud environment. The main goal of the proposed Multi-Attribute Combinative Double Auction (MACDA) resource allocation is for the efficient resource provisioning using auction based technique to fulfill the expectations of both customers and providers in cloud environment. The experimental results clearly illustrated that the proposed MACDA model was cost effective, efficient and intensive for both cloud customers and providers satisfaction.

6. REFERENCES

[1] GauravBaranwal, DeoPrakashVidyarthi, A fair multi-attribute combinatorial double auction model for resource allocation in cloud computing, The Journal of Systems and Software , Vol. 108, (2015), pp. 60-76.

[2] Sharrukh Zaman, Daniel Grosu, Combinatorial auction-based allocation of virtual machine instances in clouds, Parallel Distributed Computing, Vol. 73, (2013), pp. 495–508.

[3] Ruichun Tang, Yuanzhen Yue, Xiangqian Ding, Yue Qiu, Credibility-based cloud media resource allocation algorithm, Journal of Network and Computer Applications, Vol. 46, (2014), pp. 315- 321.

[4] Weiwei Kong, Yang Lei, Jing Ma, Virtual machine resource scheduling algorithm for cloud computing based on auction mechanism, Elsevier, Optik, vol.127, (2016), pp.5099–5104.

[5] Seyedeh Aso Tafsiri, Saleh Yousef, Combinatorial double auction-based resource allocation mechanism in cloud computing market, Elsevier, The Journal of Systems and Software 137 (2018), pp. 322–334.

[6] Parnia Samimi, Youness Teimouri, Muriati Mukhtar, A Combinatorial double auction resource allocation model in cloud computing, Elsevier, Information Sciences 357 (2016), pp. 201-216.

[7] BoYang, Zhiyong Li, Shilong Jiang, Keqin Li, Envy-free auction mechanism for VM pricing and allocation in clouds, Elsevier, Future Generation Computer Systems, Vol. 86, (2018), pp. 680-693.

[8] Jixian Zhang, Ning Xie, Xuejie Zhang, Weidong Li, An online auction mechanism for cloud computing resource allocation and pricing based on user evaluation and cost, Elsevier, Future Generation Computer Systems, Vol. 89, (2018), pp. 286-299.

[9] Rajkumar Buyya., Saurabh Kumar Garg., and Rodrigo N. Calheiros, 'SLA-Oriented Resource Provisioning for Cloud Computing: Challenges, Architecture, and Solutions', International Conference on Cloud and Service Computing, (2011), pp. 1-10.