ASK-BID: A Nash Auction Equilibrium Model for Service Provisioning in the Multi-Cloud Environment

Mrs.K.Chitra and P.Siva Prakasam

Research Scholar, Department of Computer Science, Mother Teresa Women’s University, Kodaikanal, Tamilnadu India

Abstract

The growth of well-organized service provisioning strategies becomes progressively more challenging. In the multi-cloud environment, several cloud providers are available to offer their services based on the different cost. Many existing auction schemes are mainly focused on the profit maximization of service providers. The optimal cost of user-defined service prediction is not functioning properly. To overcome these existing problems, the ASK-BID mechanism is proposed in this paper. The optimal reservation cost is implemented using Nash equilibrium, which matches the value of the ASK-BID. The minimal cost of resources utilizes the maximum cloud profit and the optimal time for cloud users. The negotiation of the service and its cost is rely upon the time slot in which it provides the continuous fluctuations in their states. This uncertainty and the cost of service migration due to the uncertainty is complex to presage. The novel approach is introduced to predict and solve the uncertainty of the cloud services depends on forecasting their load computation. An optimal cost computation mechanism is presented to estimate the initial auction payment, and utilization period payment. Subsequently, the minimal cost and minimal time service provider can be forecasted by the proposed approach in the multi-cloud environment. The experimental result exhibit better execution time, optimal cost, resource utilization and user satisfaction than the existing method.

Introduction

Cloud computing is an emerging technology in which the massive amount of computing resources is dispersed over the virtual datacenter. It can be approached as the service through user interfaces such as the web browser. As per the requirement via the internet, the pool of computing resources are made available to the user. The computing resources are allocated to the cloud user for executing their tasks. One of the computing services is an Infrastructure as a Service (IaaS), which exploited in the cloud environment. The cloud user requires virtualization technologies to access the cloud resources for IaaS environment.

Cloud provisioning allocates the resources of cloud providers to a customer. It must create the appropriate number of virtual machines (VMs) once a cloud provider accepts a request from the cloud user. This process is conducted in the advance provisioning, dynamic provisioning, and user-self provisioning. In the multi-cloud architectures, there are different architectural models for cloud service provisioning (Ferrer, A., et al., 2012). The tasks of cloud provisioning are accomplished by the cloud users. Those cloud users are the members of virtual provisioning groups. The members of these groups are involved in all required tasks among the cloud provisioning. The workflow of the cloud service provisioning is shown in Fig.1 and the members are:

- Cloud User
- Cloud Operator
- Cloud Administrator

The cloud user can request VMs from the service and employ the virtual portal for managing the assigned VMs. The cloud operator fulfills provisioning requests from the cloud users and performs the day-to-day cloud provisioning works. It completes the task that arrived in its portal. The cloud administrator owns the cloud provisioning environment to configure the numerous virtualization providers. The service catalogs can be created by the cloud administrator for monitoring the cloud provisioning environment using the portal.
computing is the service-oriented modern platform, which provide the cloud services by the cloud provider on a pay-per-use basis. Many cloud providers are available to suggest their services based on the various cost. The service providers profit maximization is discussed in many existing auction mechanisms. The minimal cost of resource uses the maximum cloud profit and the optimal time for cloud users. The cloud users increase the user’s profit and the time for the resource utilization in which it is easily accessed. The optimal cost for user service prediction is not working in the accurate manner.

![Diagram](image)

**Fig. 1:** The workflow of cloud service provisioning.

Therefore, a novel ASK-BID mechanism is proposed to overcome the above existing problem. Using Nash equilibrium concepts, the optimal reservation cost is implemented in the proposed scheme. The negotiation of the service and their cost depends on the time slot is provide the continuous changes in their states. To predict this uncertainty and the prediction of cost for migration of the service due to the uncertainty is complex. The novel approach is present to forecast and solve the uncertainty of the cloud service based on their load computation. Moreover, an optimal cost computation mechanism is employed to forecast the initial auction payment and the utilization period payment. From this approach, we can able to predict the service provider’s minimal cost and minimal time in the multi cloud environment.

This paper is structured as follows. Section I reviews the related work on the concept of resource provisioning using auction mechanism, Nash equilibrium, service based schemes in the cloud computing environment. Section II describes the overall explanation of the proposed methodology. Section III evaluates the performance of the proposed method with the existing technique. Section IV summarizes with a brief conclusive remark and discussion on future works.

**I. Related Work:**

This section deals with the related work based on the resource provisioning using auction mechanisms, Nash equilibrium and service-based application in the cloud computing environment. Kousiouris et al presented a two-level generic black box model for managing the behavior of the user across the cloud services. The benefits of this model were: (a) necessitates resource provision, (b) minimizes the cost and guarantees the QoS provision, (c) reduces the time, (d) enhance the capacity of infrastructure, and (e) limits the amount of information across cloud stack layers (Kousiouris, G., et al., 2014). Anselmi et al proposed a game theoretic approach, which model the problem of generalized Nash equilibrium. This approach takes the SaaS provider’s perspective that host its application at a PaaS provider. The revenues were determined and attains the performance level by maximizing the profit for each SaaS provider (Anselmi, I., et al., 2014). Tang et al considered a multi-tier setting, which designs a hierarchical auction mechanism along with the desired properties. It was designed for network resource allocation, where each sub-models were either a first-cost or second-cost auction at tier 1. The efficient Nash equilibrium was implemented for the single-sided and the double-sided auctions (Tang, W., and R. Jain, 2012).

Ardagna et al introduced the service provisioning problem model as a generalized Nash game. Based on the best reply dynamics, two solution methods were presented. To generalize Nash equilibrium, it requires to prove their convergences in a finite number of iterations. For the run-time allocation of IaaS resource, an efficient distributed algorithm was developed (Ardagna, D., et al., 2012). Zhang et al provided a description of both workload and machine heterogeneity that establish in clusters. A heterogeneity-aware framework (Harmony), which adjusts the number of machines dynamically. The cost was reconfigured for yielding the large energy savings and enhancing the task scheduling delay (Zhang, Q., et al.). Chaisiri et al described an optimal cloud resource provisioning (OCR) algorithm. The multi cloud providers offer the resource provision and the optimal solution was acquired by stochastic integer programming. OCR problem was divided into sub-problems by
applying Benders decomposition approach. This algorithm acts as a resource provisioning tool that effectively save the total cost (Chaisiri, S., et al., 2012).

Islam et al provided an evolutionary approach for adaptive resource provisioning in the cloud. This construct an effective prediction model for facilitating dynamic and proactive resource management. It accomplishes an on-demand resource allocation in the cloud and adaptive resource management (Islam, S., et al., 2012). Bhavani and Guruprasad studied the resource provisioning mechanism to ensure the guaranteed performance for applications. It was the challenging problem in the cloud computing environments. Static and dynamic resource provisioning requisite the parameters of Quality of Service (QoS) such as availability, throughput, reliability, and more. It avoids the violation of Service Level Agreement (SLA) (Nagesh, B. B., 2014). Trivedi et al addressed the significant novel scheduling and resource provisioning problem on IaaS clouds. This increases the number of user-prioritized tasks within the constraints of budget and deadline. It models the data access as a portion of the task execution time and does not determine the data storage and transfer cost (Trivedi, N. S. and D. S. Chudasama, 2013).

Chen et al studied the components in MapReduce processing that make decision of resource provisioning in public clouds. The Cloud Resource Provisioning (CRESP) based on the specialized MapReduce time cost model. It has a number of model parameters for a particular application, which can be learned with small clusters and small sample datasets. The optimal setting for resources can be found by the user based on the time-cost model (Chen, K., et al., 2014). Hu et al addressed the difficulties of resource provisioning and replica placement for cloud based Content Delivery Networks (CDNs). There were several algorithms for solving the long-term provisioning, short term caching, and caching problem. The total rental cost can be decreased using CNS Differential Provisioning and Caching (DPC) algorithm. The replica placement can be adjusted by Caching and Request Balancing (CRB) algorithm with DPC, which increases the runtime (Hu, M., et al., 2013). Zhou et al contributed a transformation based workflow optimization system. In real cloud environments, the workflow optimization system was developed and deployed. This addressed the performance and optimizes the monetary cost in the cloud (Zhou, A. C. and B. He, 2014). Zhu et al developed a feedback control based approach that increases the application QoS. This framework was evaluated with two real adaptive applications. Through the use of the resource model, the CPU cycle per memory allocation was generated. It yields a time constraint and a resource budget limit. The dynamic resource provisioning algorithm attains lees overhead via a static provisioning scheme (Qian, Z. and G. Agrawal, 2012).

Zhan et al proposed a co-operative resource provisioning solution for saving the server cost. The server cost is considered as the biggest share of hosting data center costs. For enabling co-operative resource provisioning for heterogeneous workloads, Phoenix cloud was built. This saves the minimum server cost with regards to the non-cooperative systems (Zhan, J., et al., 2013). Buyya et al evaluated an energy-aware resource allocation algorithm, which utilizes the dynamic consolidation of virtual machines (VMs). It minimizes the energy consumption in cloud data centers when compared to the static resource allocation. Also, improve the energy efficient management in the cloud computing environment (Beloglazov, A., et al., 2012). Shi et al presented a cloud Continuous Double Auction (CDA) mechanism for cloud resource allocation in which the market rules were defined for matching orders. An electronic bidding platform was designed for implementing the CDA mechanism in the cloud environment. A two stage game bidding strategy was developed for cloud depends on the improved belief function. It provides better performance and enhances the market efficiency (Shi, X., et al., 2013).

Zaman et al addressed the problem of VM provisioning and allocation. This was designed using a combinatorial auction-based mechanism. It was based on the user validation and produces an efficient resource allocation and high profits for the cloud provider (Zaman, S. and D. Grosu, 2013). Misra et al identified the bandwidth shifting and redistribution problems in the mobile cloud computing environment. The problem of bandwidth redistribution varies from the problem of traditional bandwidth allocation. An auction based QoS guaranteed utility maximization algorithm was introduced. This algorithm maximizes the revenue and maintains QoS of mobile nodes (Misra, S., et al., 2013). Buyya et al proposed a routing and scheduling algorithm for the cloud architecture. This algorithm minimizes the total energy consumption and the energy model makes a single step decision. The resource provisioning was calculated in a two step approach for comparing the one step scheduling with the traditional scheduling schemes (Buyysse, J., et al., 2013). Nallur et al presented a decentralized mechanism for self adaptation via the market based heuristics. A continuous double-auction was employed to permit applications for deciding the services among several services (Nallur, V. and R. Bahsoon, 2013).

II. A Nash Auction Equilibrium (NAE) Model:

This section describes the overall explanation of a Nash auction equilibrium model for the service provisioning in the multi cloud environment. The resource model depends on a set \( R \) of heterogeneous resources in which each \( r \) belongs to a resource provider. The resources are accessed based on the pay-as-you-gobasis. For each individual atomic resource, the costs are established dynamically through the auctions initiated by the
providers. The overall workflow of the proposed methodologies is depicted in fig.2. The process of resource allocation is explained with the following phases:

- An auction is created by the resource manager for providing a resource and builds an initial cost.
- By submitting bids to the auction, the users attempt to win access to the resource.
- Asks are submitted to the auction by the resource manager and maximizes the cost.
- A match should be detected among the lowest value of ask and the highest value of the bid. If the user submitted the highest bid value, then the auction is won.
- The auction is completed without a winner, if no match occurs within the established time limit.
- The winner of the auction is notified by the resource manager.

![Fig. 2: An overall flow diagram of a Nash Auction Equilibrium (NAE) model for service provisioning in the multi cloud environment.](image)

Initially, the multi-cloud environment is created with the cloud configurations such as brokers, data centers, and virtual machines (VMs). The VMs are set up with the RAM, bandwidth, and CPUs. The profit for the end-user are maximized by the scheduler and resource manager. The cheap and fast resources are accomplished from the scheduler, whereas the resource manager increases its revenue or resource utilization and also ensures the user fairness. The above negotiation process results in a final set of resources. These resources are provisioned for the application execution. After creating the VM, it is further connected to the broker and the reservation plan is established. To publish the reservation plan, (a) select either an execution service or storage service, (b) select the plan, and (c) estimate the cost. The end-user or a prediction service manually indicates the execution time of activities and data transfers on an available individual resource. These resources are accessible as a prediction matrix by a prediction service. The user gets the requested application after publishing the reservation plan. At that time, the values of bid and ask are estimated.
Table I: Description of the used variables in the Proposed Algorithm.

<table>
<thead>
<tr>
<th>S.NO.</th>
<th>VARIABLE</th>
<th>FOR BID VALUE COMPUTATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>𝐶_𝑗</td>
<td>Cost for jᵗʰ job</td>
</tr>
<tr>
<td>2.</td>
<td>𝐻𝐶_𝑗</td>
<td>Highest Cost for jᵗʰ job</td>
</tr>
<tr>
<td>3.</td>
<td>𝜑_𝑗</td>
<td>Highest Deadline</td>
</tr>
<tr>
<td>4.</td>
<td>𝜇_𝑗</td>
<td>Deadline for jᵗʰ job</td>
</tr>
<tr>
<td>5.</td>
<td>𝛼</td>
<td>Number of processor required for jᵗʰ job</td>
</tr>
<tr>
<td>6.</td>
<td>𝛽</td>
<td>Number of processor supplied for jᵗʰ job</td>
</tr>
<tr>
<td>7.</td>
<td>𝜏</td>
<td>Current Time</td>
</tr>
<tr>
<td>8.</td>
<td>𝛾_𝑗</td>
<td>Submission Time for jᵗʰ job</td>
</tr>
<tr>
<td>9.</td>
<td>𝜗_𝑗</td>
<td>Minimum Submission Time for jᵗʰ job</td>
</tr>
</tbody>
</table>

FOR ASK VALUE COMPUTATION

<table>
<thead>
<tr>
<th>S.NO.</th>
<th>VARIABLE</th>
<th>FOR ASK VALUE COMPUTATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.</td>
<td>𝛿_𝑖</td>
<td>Expected Waiting Time</td>
</tr>
<tr>
<td>11.</td>
<td>𝐶_𝑖</td>
<td>Cost for iᵗʰ VM</td>
</tr>
<tr>
<td>12.</td>
<td>𝐻𝐶_𝑖</td>
<td>Highest Cost for iᵗʰ VM</td>
</tr>
<tr>
<td>13.</td>
<td>𝜕_𝑖</td>
<td>Load for iᵗʰ VM</td>
</tr>
<tr>
<td>14.</td>
<td>𝛿_𝑖</td>
<td>Maximum Load for iᵗʰ VM</td>
</tr>
<tr>
<td>15.</td>
<td>𝛼</td>
<td>Required number of processor for the iᵗʰ job</td>
</tr>
<tr>
<td>16.</td>
<td>𝛽</td>
<td>Available number of processor for the iᵗʰ job</td>
</tr>
<tr>
<td>17.</td>
<td>𝛾_𝑖</td>
<td>Expected execution time for kᵗʰ job in iᵗʰ VM which is already allocated in the for the iᵗʰ VM</td>
</tr>
<tr>
<td>18.</td>
<td>𝜔_𝑖</td>
<td>Size of kᵗʰ job in MI</td>
</tr>
<tr>
<td>19.</td>
<td>𝜌_𝑖</td>
<td>Speed of iᵗʰ VM in MIPS</td>
</tr>
</tbody>
</table>

FOR INITIAL COST COMPUTATION

<table>
<thead>
<tr>
<th>S.NO.</th>
<th>VARIABLE</th>
<th>FOR INITIAL COST COMPUTATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.</td>
<td>𝐶_𝑖_𝑗</td>
<td>Unit Cost for jᵗʰ job in iᵗʰ VM</td>
</tr>
<tr>
<td>21.</td>
<td>𝜔_𝑖_𝑗</td>
<td>Unit Cost in iᵗʰ VM</td>
</tr>
</tbody>
</table>

FOR UTILIZATION COST ESTIMATION

<table>
<thead>
<tr>
<th>S.NO.</th>
<th>VARIABLE</th>
<th>FOR UTILIZATION COST ESTIMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.</td>
<td>𝐴_𝑖_𝑗</td>
<td>Utilization Cost for jᵗʰ job in iᵗʰ VM</td>
</tr>
<tr>
<td>23.</td>
<td>𝜔_𝑖_𝑗</td>
<td>Unit Cost in iᵗʰ VM</td>
</tr>
</tbody>
</table>

A. **BID Value Computation:**

In each time unit, the user agent defines a value of bid based on the parameters of the job cost, deadline, and the average remaining time. Table I describes the used variables in the following equations. The bid value computation can be obtained by the equation (1).

\[
\text{bid} = \frac{C_j}{HC_j} \cdot \frac{(\phi_j - \tau)}{(\mu_j - \tau)} \cdot \frac{\alpha}{\beta} \cdot (\gamma_j + t) 
\]

B. **ASK Value Computation:**

The server agent determines the value of ask, where the resource obtains highest profit at the moment of linking the multi-cloud environment. For accepting a task, the initial time of resource is 0 and the cost is set as the reserve cost. These are further updated to the maximum cost after accepting a task. Therefore, the initial time and cost of resource are minimized gradually. The computation of ask value is as in equation (2):

\[
\text{ask}_i = \frac{\Delta_i}{HC_i} \cdot \frac{C_i}{\tilde{C}_i} \cdot \frac{\delta_i}{\tilde{\delta}_i} \cdot \frac{\alpha}{\beta} \cdot (\gamma_i + t) 
\]

Here, the expected waiting time is the summation of the expected execution time for a job in the VM, which is already allocated in the VM. It is mathematically represented as follows:

\[
\Delta_i = \sum_{k=0}^{\infty} \theta_{ki} 
\]

The equation (3) is acquired by applying the equation (4) with the job size and the speed of the VM.

\[
\theta_{ki} = \frac{\sigma_k}{\rho_i} 
\]

C. **Nash Auction Equilibrium:**

A match is detected between the lowest ask value and the highest bid value using Nash Auction Equilibrium (NAE).

The proposed algorithm is reached as a non-cooperative game, which increases the average and economic performance of the Quality of Service (QoS). The inputs used in the algorithm are the list of virtual machines...
and the job list, and the output is the allocated job list. At first, the allocated job list is set as empty and the user agent determines their bid value using the equation (1). Initialize the ask matrix as empty and the ask value is computed by the equation (2). A match is detected using Nash equilibrium and the corresponding output is obtained if a match occurs.

### ASK-BID: A Nash Auction Equilibrium Algorithm

**Input:** $V_{List}$, $J_{List}$  
**Output:** $AJ_{List}$  

1. $AJ_{List} = \emptyset$ // Initialize allocated job list  
2. $\mathcal{G}_j = \text{bid}_j$ from Equation 1  
3. $\mathcal{A} = \emptyset$ // Initialize Ask Matrix  
4. $\mathcal{A}_i = \text{ask}_ij$ from Equation 2  
5. For $I = 0$ to $M$ then  
6. Match $\mathcal{G}_j$ with $\mathcal{A}_i$ // Using Nash Equilibrium  
7. If (Match == true)  
   - $AJ_{List}J = \mathcal{A}_i$  
   - Break  
8. End If  
9. End For  
10. End For  

**D. Initial Cost Calculation:**  

Once the winner notification is acquired, an initial provision cost is performed. The initial cost is calculated as the product of the job size and the VMs unit cost.  

$$C_{ij} = \sigma_j \cdot \omega_i$$  

(5)

**E. Utilization Cost:**  

By simulation or utilization, the cost utilization is computed for the proposed system. The utilization cost is termed as the product of the job execution time and the unit cost in the VM and is equated as follows:  

$$UC_{ij} = \theta_j \cdot \omega_i$$  

(6)

**F. Overall Cost:**  

Finally, the optimal cost is estimated by the product of the equation (5) and the equation (6).  

$$OC_{ij} = C_{ij} \cdot UC_{ij}$$  

(7)

The performance analysis is evaluated for the proposed method in the next section.

### III. Performance Analysis:

This section analyzes the performance of cost optimization, resource utilization, user satisfaction, execution time in the multi-cloud environment with regards to the number of bidders, load size, number of VM, makespan time. For modelling the framework of a cloud computing environment, a CloudSim (Cloud Simulator) tool is used for executing the proposed algorithm. In the simulation, users and servers are modelled as two kinds of agents. The cloud configures the brokers, datacenters, and the virtual machines. Table II determines the configuration parameters used for the performance analysis.

The multi-cloud environment is created by the topology files and the number of links. After creating the multi-cloud environment, the datacenter is created and then the VM is established. Further, it connects to the broker and the reservation plan for cloud service provider is proceeding. The plan is published by selecting either executive or storage service, selecting the plan, and confirming the cost per MI (Million Instruction). The cloudlet and reservation requests are generated to calculate the bid and ask values. A match is detected using Nash equilibrium and the results are produced for matched values. The provision cost is computed after attaining the results and the simulation gets started for utilization cost.

<table>
<thead>
<tr>
<th>PARAMETERS</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>10</td>
</tr>
<tr>
<td>Number of Jobs</td>
<td>50 to 100</td>
</tr>
<tr>
<td>MIPS</td>
<td>1500 to 4000</td>
</tr>
<tr>
<td>Bandwidth (BW)</td>
<td>1000 to 4000</td>
</tr>
</tbody>
</table>
A. Load Size vs. Resource Utilization:

The resource utilization is the ratio of the number of allocated resources to the total number of resources. Fig. 3 depicts the result of resource utilization with respect to the load size for the existing Vickery-Clarke-Groves (VCG) Auction (Zhang, L., et al., 2014) approach and the proposed Nash Auction Equilibrium (NAE). This provides high resource utilization when compared with the existing mechanism.

![Resource Utilization Graph](image)

Fig. 3: The result of resource utilization with respect to the load size for the existing VCG auction and the proposed NAE.

B. Number of Bidders vs. Payment:

Fig. 4 depicts the result of payment with respect to the number of bidders for the existing Vickery-Clarke-Groves (VCG) Auction (Zhang, L., et al., 2014) approach and the proposed Nash Auction Equilibrium (NAE). The payment for each bidder is lower than the existing approach.

![Payment Graph](image)

Fig. 4: The result of payment with respect to the number of bidders for the existing VCG auction and the proposed NAE.

C. Number of Bidders vs. User Satisfaction:

The user satisfaction is the ratio of the number of winning auction to the total number of auctions. Fig. 5 shows the result of user satisfaction with respect to the number of bidders for the existing Vickery-Clarke-Groves (VCG) Auction (Zhang, L., et al., 2014) approach and the proposed Nash Auction Equilibrium (NAE). The proposed NAE provide better user satisfaction than the existing approach, which gradually decrease its value.
Fig. 5: The result of user satisfaction with respect to the number of bidders for the existing VCG auction and the proposed NAE.

D. Number of VM vs. Execution Time:

For each virtual machine, the performance of the proposed Nash Auction Equilibrium (NAE) in terms of the execution time is compared to the existing Vickery-Clarke-Groves (VCG) Auction (Zhang, L., et al., 2014) approach. The proposed NAE offers the minimum execution time than the existing approach as depicted in fig. 6.

Fig. 6: The result of execution time with respect to the number of VM for the existing VCG auction and the proposed NAE.

E. Load Size vs. Execution Time:

Fig. 7 illustrates the result of execution time in terms of the load size for the existing Vickery-Clarke-Groves (VCG) Auction (Zhang, L., et al., 2014) approach and the proposed Nash Auction Equilibrium (NAE). The NAE provides a better execution time when compared to the existing mechanism, which gradually increases its value for each load size.

Fig. 7: The result of execution time with respect to the load size for the existing VCG auction and the proposed NAE.
F. Makespan Time vs. Optimal Cost:

The optimal cost is the profit-maximizing cost, where marginal revenue is similar to the marginal cost. Fig. 8 presents the result of optimal cost with respect to the makespan time for the existing Vickery-Clarke-Groves (VCG) Auction (Zhang, L., et al., 2014) approach and the proposed Nash Auction Equilibrium (NAE). The proposed model provides minimum optimal cost than the existing approach in which its values are gradually decreased for each makespan time.

![Fig. 8: The result of optimal cost with respect to the makespan time for the existing VCG auction and the proposed NAE.](image)

IV. Conclusion and Future Work:

Focusing on service provisioning, we propose a Nash auction equilibrium model to implement the optimal reservation cost. A novel approach is introduced for predicting and solving the uncertainty of the cloud services based on their load computation. To estimate the initial auction payment and the utilization payment period, an optimal cost computation mechanism is presented. Successively, the minimal cost and minimal time service provider can be forecasted in the multi-cloud environment. Simulation studies outperform the existing VCG auction approach in terms of the execution time, optimal cost, and resource utilization, user satisfaction.

In future, the proposed methodology will enhance the workflow job based e-auction approach by considering the energy and the load. It will accomplish the minimum cost, less fault value, and minimum job execution rate.

REFERENCES


