Alfarama Journal of Basic & Applied Sciences

https://ajbas.journals.ekb.eg
ajbas@sci.psu.edu.eg

Faculty of Science Port Said University

http://sci.psu.edu.eg/en/

DOI:https://doi.org/10.21608/ajbas.2023.203105.1152

October 2023, Volume 4 , Issue IV

ISSN 2682-275X

Submitted: 09/04/2023
Accepted: 19/07/2023

Pages: 635 - 649

Design and Implementation an Intelligent Dynamic Negotiation with Third Party for Cloud Computing

D.M. El-Matary[1,*], N. E. El-Attar [2], W. A. Awad[3] and I.M.Hanafy [4]

[1] Industrial Technical Institute ,Technical college, Port Said ,Egypt

[2] Faculty of Computers and Artificial Intelligence, Benha University, Egypt.

[3] Faculty of Computers and Information, Damietta university, Egypt

[4] Faculty of Science, Portsaid University, Portsaid, Egypt

***Corresponding author: dodomomo0122@gmail.com**

ABSTRACT

Only a successful network connection is required for the cloud computing concept to provide access to information and computing resources from anywhere, to keep up with the dynamic aspects of the cloud environment, such as multiple distributed systems and multi-tenancy, since cloud computing is inherently multi-tenant, complex, large-scale, and heterogeneous. So, it needs to automate and integrate its operations and strategies, respectively. The Service Level Agreement (SLA) is a formal negotiated agreement that helps to identify expectations, clarify responsibilities, and facilitate communication between the service provider and the users. This paper aims to create a framework for dynamic service level agreements, which always have to be elastic and flexible in handling and translating the user services' requirements. Also proposed negotiation framework primarily relies on intelligent agents that play the role of third parties to overcome obstacles in static negotiations, like the ongoing changes in business service requirements.

Key Words:

Cloud Computing, Service Negotiation (SLA), Dynamic negotiation.

1. INTRODUCTION

The definition of cloud computing is a new paradigm for on-demand hosting services delivery over the Internet. A cloud provider offers pay-per-use internet-based services that include resources (such hardware, software, or development stacks) [1]. A user of cloud computing can access these programmers and services via a browser regardless of where they are located [2]. The user can utilize the Cloud infrastructure by using the Cloud Computing apps without wasting time installing and maintaining any components on his machine[3].

Cloud computing delivers infrastructure, platform, and software (applications). In industry, these services are referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as Service

(SaaS). On the other hand, the Cloud aims to power the next generation of data centers by architecting them as a network of virtual services [4].

Guarantee the process of service delivery is a vital requirement for the Cloud's users. This guarantee can be achieved through a negotiation called Service Level Agreement (SLA), contains the conditions and the rights of both the providers and the users [5]. Generally, The Service Level Agreement (SLA) is a formally established contract that helps in defining expectations, outlining duties, and facilitate communication between the service provider and its customers.

The SLA negotiation allows the user to access the cloud market by a high level business interface. The user can choose among the supported negotiation models. The negotiation interface allows to delegate a lot of stuff to the cloud application and to focus on the desired solution in terms of requirements, service levels and business parameters [6].

This article describes a compilation of prior efforts in the case of the third party with a suggestion for a new design with an intelligent third party that makes it's tasks distinct, in an effort to address the broker's shortcomings, the rest of the paper will describe the service level agreement for cloud computing and its challenges, and then presenting earlier efforts of addressing a dynamic SLA, the proposed system frame work next, developing a platform to simulate the proposed system, and then evaluating performance.

1.1. Service Level Agreement For Cloud Computing: The Service Level Agreement (SLA) is a vital contract that exists between the service providers and the service's user [14]. The basic goal of SLA is to clearly define official agreements involving terms of service including performance, availability, and billing..., etc. [14]. SLA serves as a legal template for recording the method in which services are provided between vendors and customers, and everyone will deal with the SLA from his point of view. From the perspective of the service provider, he uses the contract to optimize the infrastructure While, the Service's user deals with the SLA to ensure the required level of service quality and to maintain acceptable business models for long-term provisioning of services [15].

When signatory parties (user and provider) to a Service Level Agreement (SLA) decide to contract specific tasks [8], a third party (broker) enters the picture as seen in Figure 1. The broker is an important part of managing how a cloud service is delivered. In addition, a cloud broker can help in the role of a mediator between service providers and clients [8], where the third party is thought of as an organization that facilitates secure interactions between two parties. The broker receives the user request and figures out the anticipated cost, anticipated refresh time, anticipated processing time, and availability [12].

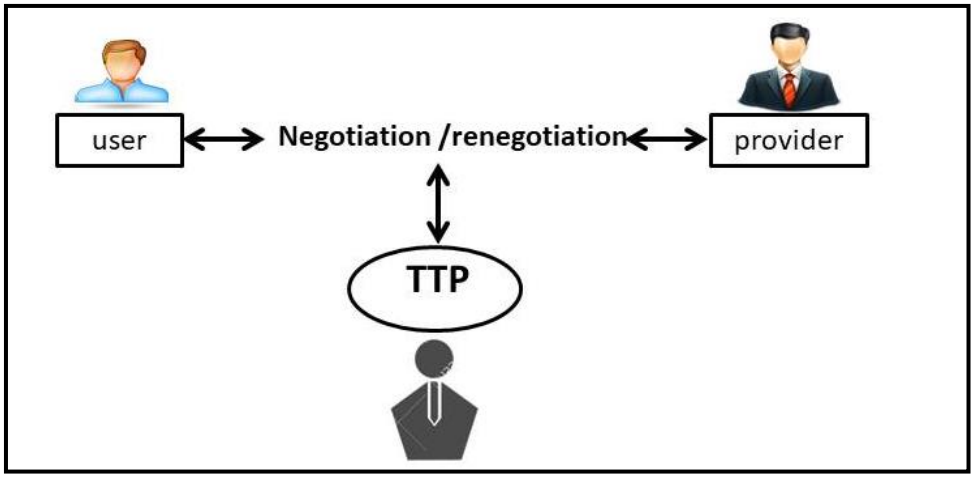

Fig (1) : the main roll of trusted third party, which act as a mediator between service providers and users

1.2. Challenges in Service Level Agreement :

1- SLAs are necessary for performance-critical applications so that the computing infrastructure can monitor, identify, and react to changes in demand and meet application-level processing needs.
2- Demand changes may result from the operating apps themselves and also from the creation of new applications.
3- Applications may have unpredictable changes in their processing demands and associated service levels.
4- It would be desirable for the cloud service provider to be able to adapt to demand changes even if they are relatively predictable without needing to renegotiate a new SLA.
5- SLA will be particularly necessary in computing cloud that are, by nature, multi-tenant environments where many applications may have changing service level requirements.

By providing dynamic SLAs, we are attempting to satisfy competing goals: (a) ensure that every running application component meets its deadlines, while (b) enabling the cloud scheduler to maximize resource utilization, thereby "doing more with less", also applying dynamic negotiation will help avoid forcing users to over specify their service level requirements in order to satisfy future changes in their demand.

2. RELATED WORK

Zulkernine. F. and Martin. P.[7] have proposed an automated web-based negotiation method for Service Level Agreements in light of the technology of agents that has been created to deal with complex systems and improve dynamic negotiation in Cloud Computing. Their suggested web-based automated negotiation's most important feature is that it establishes the QoS needs of crucial services based on a "Trusted Negotiation Broker" framework that executes an adaptive and intelligent SLA contract between the service provider and the service consumer. Via the dependable negotiation broker, the ultimate negotiation result is implemented.

Al-aaidroos. M. et. al. in [8] have suggested a conceptual framework and a prototype implementation to achieve a fully automated web service's SLA lifecycle in a somewhat similar way. The Service Provider Module (SPM) and the Service Consumer Module (SCM), which are based on the web service standards Schema, are the two main modules that make up the proposed architecture. Three stages—pre-negotiation, in-negotiation, and post-negotiation—make up the suggested negotiation system. The negotiation parties prepare and present their demands during the pre-negotiation stage. The negotiation requests are then traded between the SPM and the SCM during the in-negotiation stage. The negotiation results are then produced during the post-negotiation stage.

For web services as well, the capability to dynamically and automatically build and manage SLAs is regarded as a crucial issue. To address this issue, Xiao Z. and colleagues in [9] developed a policy to support automatic and dynamic SLA negotiations for web services. The Negotiation Agent Factory (NAF), the core module of this proposed architecture, is in charge of receiving the suggested negotiation policies and decision-making models as input from the service provider or the service customer. Then, it incorporates these negotiation policies and decision-making models into the negotiation system to dynamically produce Negotiation Agents (NA).

Maximizing provider profit while raising customer satisfaction levels is another trend in creating a valuable SLA. In [10], Wu. L. et al. suggested an automated Cloud negotiating framework based on generating methods and decision-making while taking time and market considerations into consideration to achieve diverse goals for different parties. This negotiating framework's three primary elements are the Customer Agent (CA), Broker Coordination Agent (BCA), and Provider Agent (PA). The BCA, which stands in for the Cloud broker, is the main agent in this framework. It accepts client requests and bargains with suppliers to accomplish corporate goals. It consists of the Negotiation Engine (NE), Negotiation Policy Translator (NPT), Decision Making System (DMS), and SLA Generator, which is in charge of producing a SLA depending on the QoS, between the consumer and the supplier.

Therefore, Chen. J., et. al. in [11] have created a negotiation workflow based on a multi agent system, application interface layer, and cloud resource market. They have also established a negotiation framework for cloud computing. Resource Usage Agent, Resource Supply Agent, and Intermediary Agent make up the multi-agent system. Participants in the exchange of cloud resources view the intermediary agent as a reliable third party in the cloud computing market. The resource supply agent negotiates with the resource supply agent who is in possession of the resource in the cloud computing market. It also communicates with the market for cloud resources through the application interface layer to obtain market information.

A Trusted third party (TTP) based on multi agent systems has also been established by Maarouf. A., et al. in [12] in order to regulate the quality of the service contract between the customer and vendor. In this proposed strategy, the (TTP) plays a key role in managing and coordinating the numerous agents as well as overseeing the process (prevention, correction, and/or control) and service quality.
The "System Multi Agent for Automatic Negotiation of Quality of Services (SMAANQOS)" framework, which is based on JADE, was proposed by Bakraouy, Z., et al. in [13]. (Java Agent Development Framework). The four agents on this proposed SMAANQOS are produced and activated automatically when the platform is turned on. The platform is a multi-agent system. Additionally, a broker agent interacts with the system agents to obtain the needed information for user queries regarding the providers, chooses the best offer by evaluating availability and pricing against the best ratio, and then informs the customer about the provider with the best ratio.

According to the previous research, it is evident that the traditional third party plays a big role in the SLA negotiation. For example, it performs important duties including gathering data on the providers and users [11].
- Selecting the best provider for the needed services (e.g., one that offers the best pricing, best ratio, etc.) [9].
- Putting in place security measures to stop unauthorized use of resources [12].
- Repairing any damage and restoring the infrastructure to its pre-damage condition [10].

Table 1: Related works for dynamic service level agreement SLA.

year	Reference no	Proposed Technique	Objective	Type of Negotiation
2011	7	Web-based negotiation	Improve dynamic negotiation in Cloud Computing	Negotiation with trusted Broker.
2011	8	Framework and a prototype implementation for service's SLA .	Achieve a fully automated web service's SLA lifecycle.	Two main modules with three stages.
2011	9	Negotiation policies and decision-making models.	Support automatic and dynamic SLA negotiations for web services.	The Negotiation based on agent factory (NAF),
2013	10	Cloud negotiating framework based on generating methods and decision-making	Maximizing provider profit while raising customer satisfaction levels.	Broker and agents based negotiation
2014	11	Negotiation framework and resource market for cloud computing.	In the in order to exchange of cloud resources between the participants .	Negotiation based on a multi agent with intermediary agent.
2015	12	Trusted third party (TTP) based on multi agent systems framework.	regulate the quality of the service contract between the customer and vendor.	Negotiation with trusted Broker and agents.
2018	13	Java agent development framework for automatic negotiation.	Chooses the best offer by evaluating availability and pricing against the best ratio.	multi-agent system. Additionally, a broker agent.

3. DYNAMIC SERVICE LEVEL AGREEMENT

This section presents a complete definition for "dynamic SLA" [16], as showed in Figure 2,which is described how to deal with the change of users' requirements during run time, and how this changes have to be adjusted in the SLA.

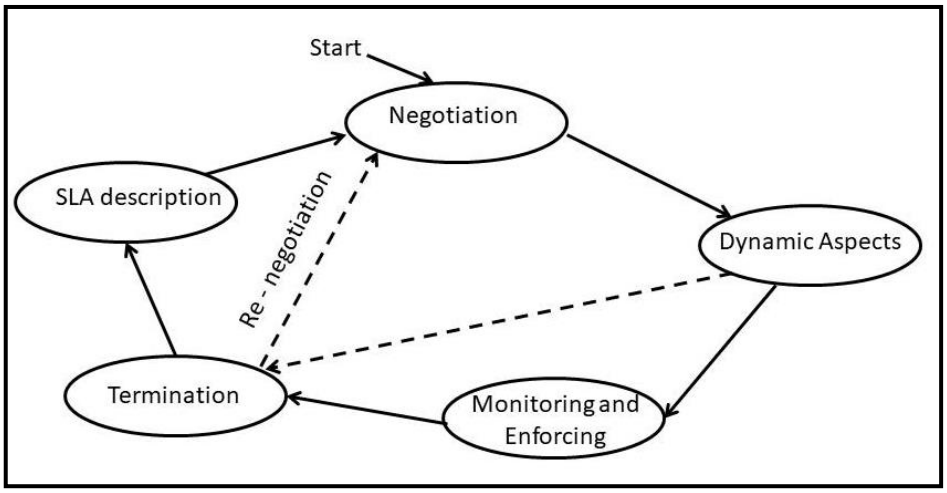

Fig (2) : life cycle for dynamic SLA, which outlined how to handle changing user requirements in real time and how these changes must be accounted for in the SLA.

1- **SLA Description**: which specify the parties involved inside the agreement defined by its name and its roles, as well as the duration (start and end times) of the validity period. Also specifies the details of the service and its quality.

2- **Negotiation:** this component represent the SLA adjustments according to user requests and provider requirements, also allows for the usage of several SLA objectives .

3- **Dynamic aspect**: the most important elements of the service level objectives, such quality and cost, might be changed at this stage in order to build renegotiation strategies.

4- **Monitoring:** the dynamic SLA feature offers process monitoring, which includes gathering data, allowing access to the negotiating parties, responding to events or requests, and getting a full view of the current system's status .

5- **Enforcement:** when violations or service changes are received, this stage calculates costs, penalties, bills of the parties and processes modifications.

6- **Termination:** SLA termination refers to the current contract's termination and ensuing renegotiation.

3.1 Intelligent Dynamic Negotiation Phases: In an intelligent dynamic negotiation, to conduct the negotiating functions in a dynamic manner, several auxiliary elements are needed. These components are structured through three basic stages, which are [17] :

1- Pre-Negotiation Phase:

This phase is considered as the first stage, which includes the beginning of the establishment of the agreement, so that the objectives are prepared on the dynamic principle, where the objectives are ready to be modified according to the different changes in the requests.

2- Negotiation Phase:

Where the negotiation process takes place at this stage, which includes the negotiation strategy, which based on the negotiation protocol between user and the service provider which is set up in accordance with a set of processes, beginning with the step of generation the proper strategy and moving on to evaluating it to obtain the ideal option using decision making, while taking into consideration the requirement for future predicting of any modifications, where the output will be the accepted model.

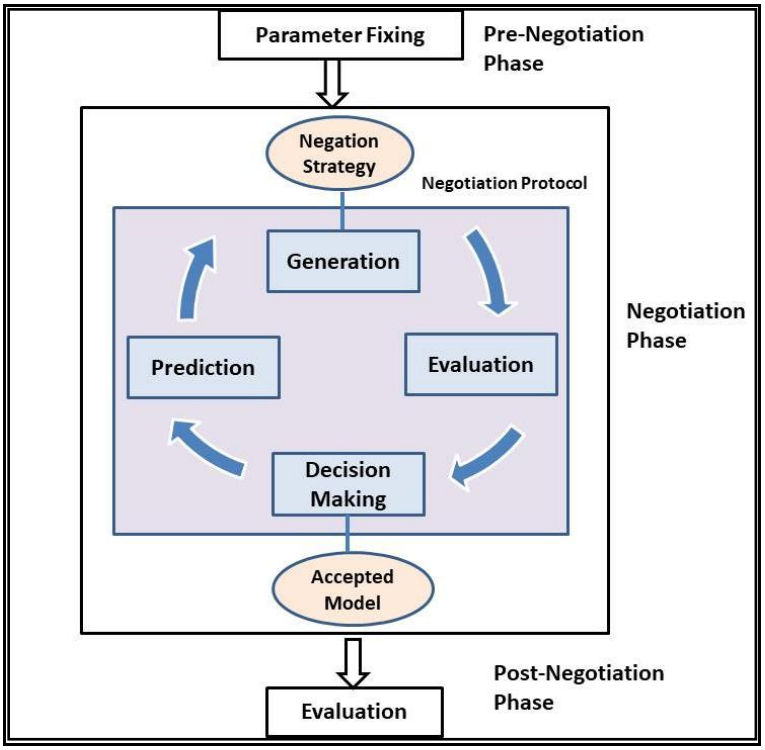

Fig (3) : the three basic stages of an intelligent dynamic negotiation 3-Post-Negotiation Phase:

This stage is considered as the last phase, where the accepted model will be evaluated to measure performance and determine the viability of the dynamic agreement. Figure 3 displays the intelligent dynamic negotiation phases.

Initially, dynamic negotiation system has been designed**,** in which an intelligent agent-based agreement paradigm is presented, where an intelligent agent acts as a third party between cloud service provider and users, where a platform was created that enables the user to modify his requests during run time with the approval of a third party, this platform was designed with our perception so that we will able to subsequently evaluate the platform's performance and accurately predict the weaknesses and challenges for any system that supports the third party.

The research idea was based on the precipitation of earlier works , and concentrated on improving the dynamic negotiation system by substituting the traditional third party with an intelligent agent which act as a third party, and then testing the performance of the dynamic system in the presence of the third party so that in future work , when creating a completely dynamic negotiation system without the third party, we will be able to make a comparison under the same conditions and using the same factors. As a result, it was used as the basis for developing a simulation system for the dynamic agreement with intelligent third party.

This approach demonstrates how we see the various phases of the negotiation parties

4. PROPOSED FRAMEWORK DESIGNE FOR INTELLIGENT DYNAMIC NEGOTIATION WITH THIRED PARTY

The proposed intelligent negotiation framework with third party . It is based on the intelligent agent as shown in Figure 4 and primarily comprises of the following fundamental elements.:

- **The User**: the service user who requests services from the service provider, and his requests are according to his application.
- **Service Provider**: offering the user cloud computing services, such as Platform as a Service (PaaS), Software as a Service (SaaS), or Infrastructure as a Service (IaaS), is the responsibility of the cloud provider.
- **Intelligent Third Party**: beside the user and the provider, there is a need to intelligent agent work as a third party which replaces the traditional third party (broker) by an agent in an attempting to overcome the common defects of the broker such as, the security issues, the repercussions of cost, risk and potential loss.
  It is role appears in the case of a change in the user resources during run time.
- **Knowledge Base** : it primarily consists of the user's information, the requested services' preferences, which will be considered as a guide during the negotiation process, and the user's submission of his service requests and QoS needs.

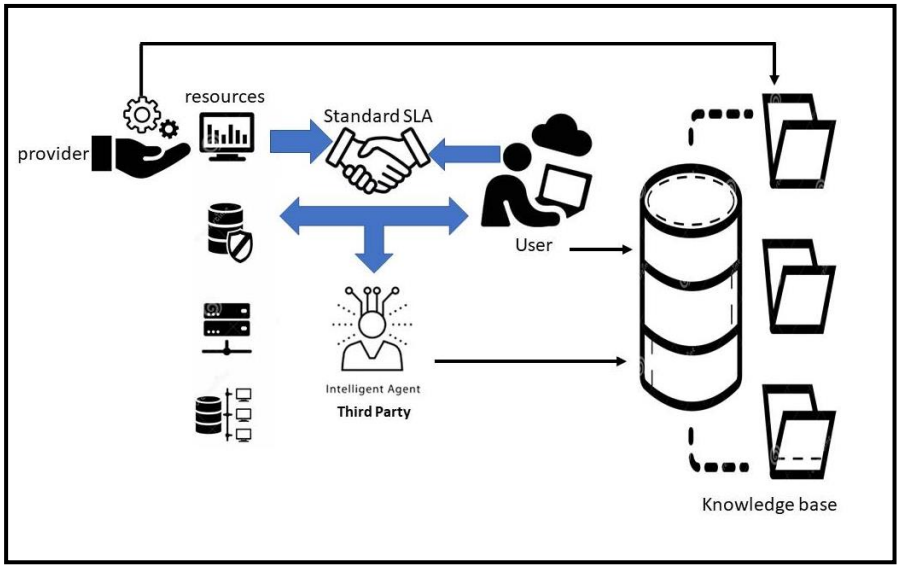

Fig (4) : intelligent dynamic negotiation with third party frame work, that include the parties of the negotiation, and the database.

The suggested dynamic negotiation method using an intelligent third party's process sequence diagram is explained in Figure 5:

1- In the suggested system, the user begins by submitting the request they made to the service provider in order to establish the agreement.
2- Depending on the unfolding conditions, the user request changes.
3- The third party studies the general conditions of the system, the resources available at the service provider, and the possibility of approving the user's requests.
4- The conditions associated with the contract between the user and the provider are renegotiated and informed, changing the agreement documentation.

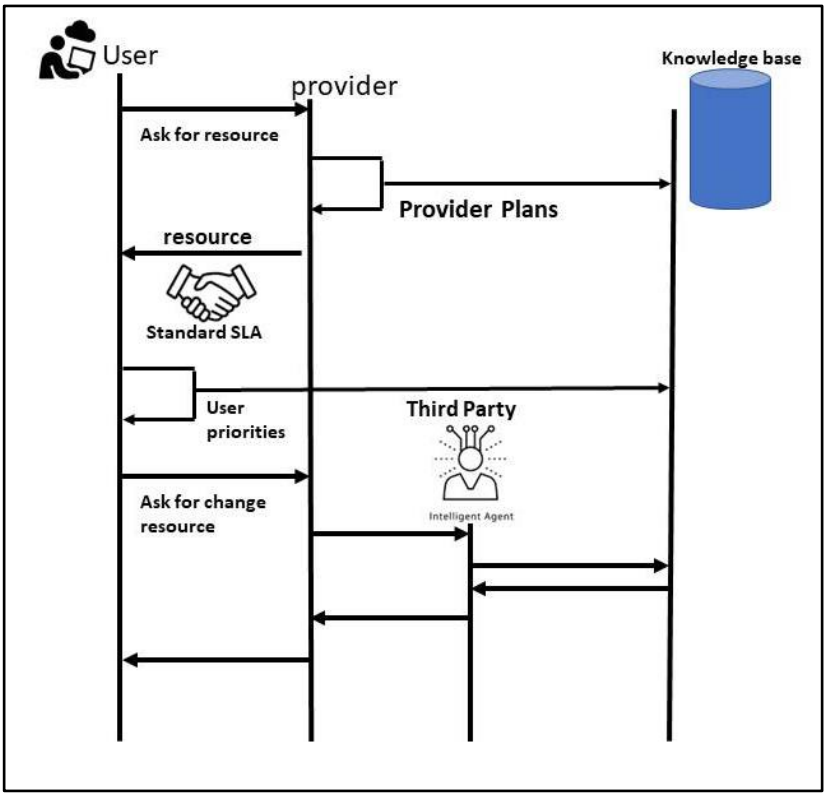

Fig (5) : the stages that the parties to the negotiations will go through are described in the frame work sequence diagram.

5. IMPELEMENTATION OF DYNAMIC PLATFORM WITH INTELLIGENT THIRED PARTY

Platform has been established which support a dynamic service negotiation, considering that the platform holds the full database mentioned in the frame work, where several service providers sign up on the platform and select the resources to be available to users. Also, the platform supports the entry of an infinite number of users, where each user requests the proper resources for his application and having the option to swap out resources until finding ones that meet his needs. Figure 6 introduced the main page window in the proposed platform, this window contains the register icons (i.e. user name, and password) for both provider and user.

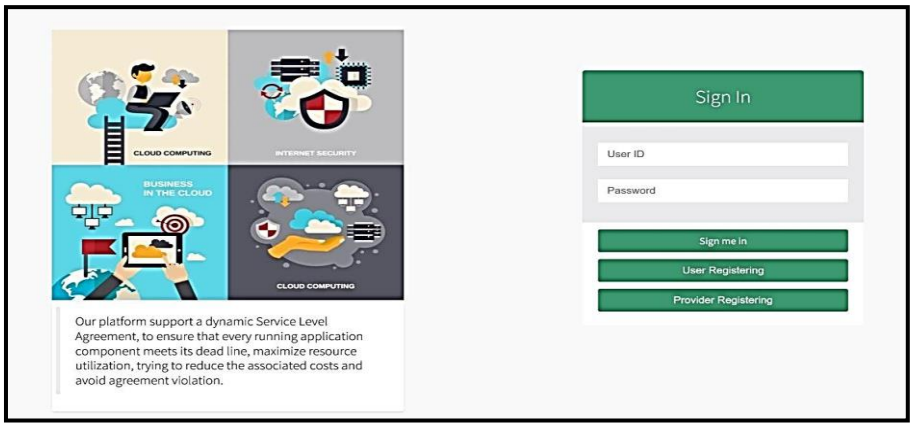

Fig (6) : proposed platform home page which display sign in window for platform users.

The provider has to fill the required information as shoewn in Figuer 7 which is divided into :

1- The basic details (e.g, . first name, last name, user name, and password).

2- Contact details (e.g. the email adress, telephone, adress).

3- The resources values that will be presented to the user (e.g.file storage, SQL database, band width, windows virsual machine).

Fig (7) : register form for new provider, that display the details that the provider has to fill.

User Process Activities: Following registration, which include only the basic and contact information, the user goes through a number of sequential processes that are represented on the platform's following pages:

- Feature Products Page

The user is free to select the values of the many sources on this page, as shown in Figure 8, each value of each resource has a clear and distinct price for the user, the platform enables the users to select the values with a cost that is suitable for their budget.

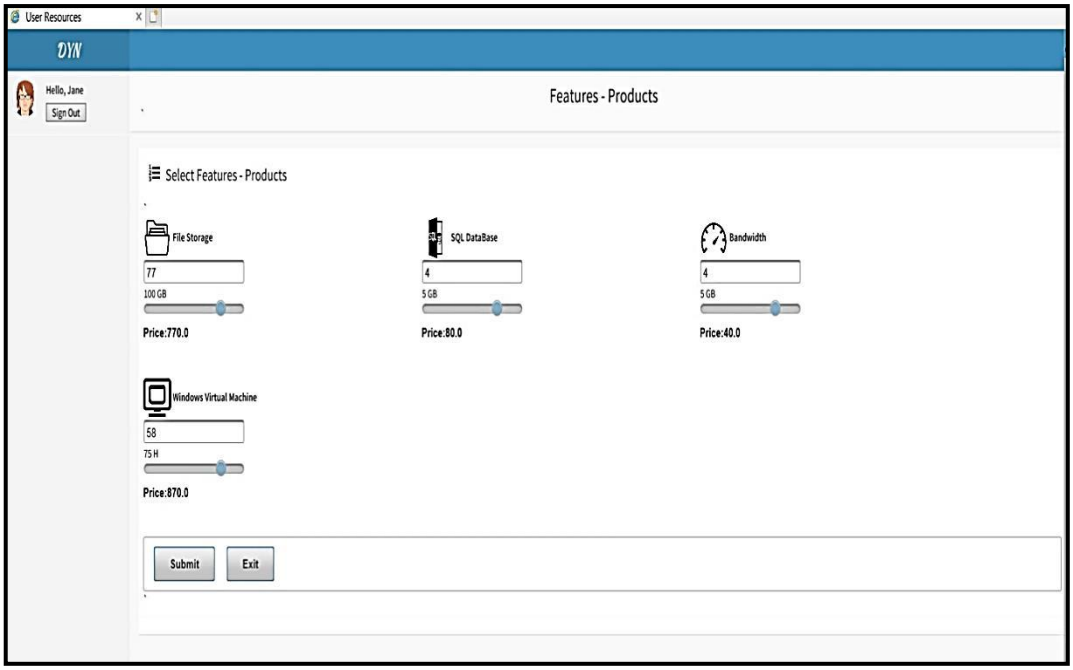

Fig (8) : user features products window, where user select the desired value

- Standard Service Level Agreement Page

It is now time to agree to the agreement on the next page shown in Figure 9, after selecting the desired resources.

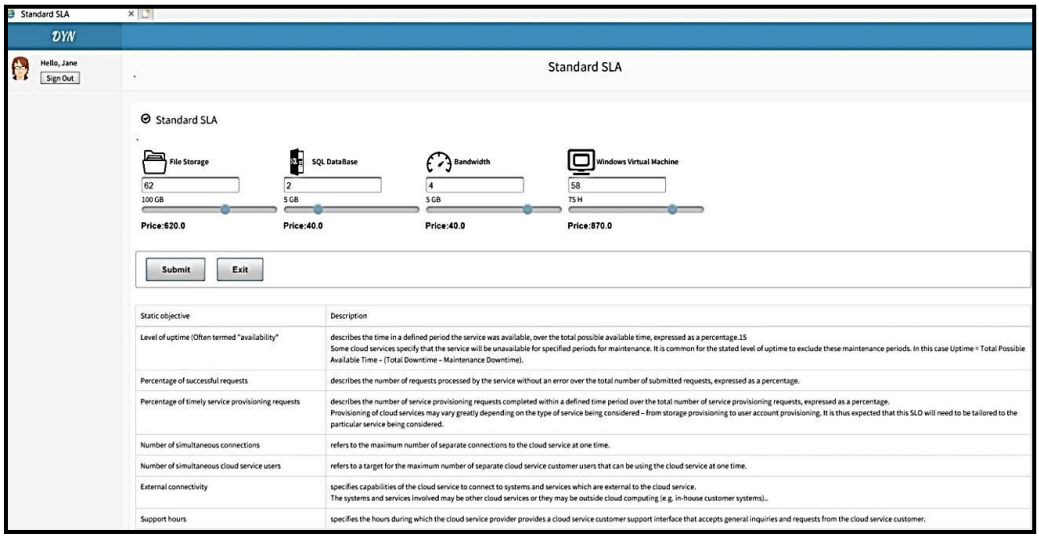

Fig (9) : standard sla window that display the agreement objectives, and its description which considered as the terms of the negotiation.

If the user decides to modify his request while it is being carried out for any circumstance, and he really changes the orders that were previously chosen, the third party will now be involved.

The role of the intelligent third party in the system displayed in Figure 10, where it has been restricted to approving the user's requests, a message will appear to the user stating that the desired request will be reviewed and so that the user must wait for a certain amount of time.

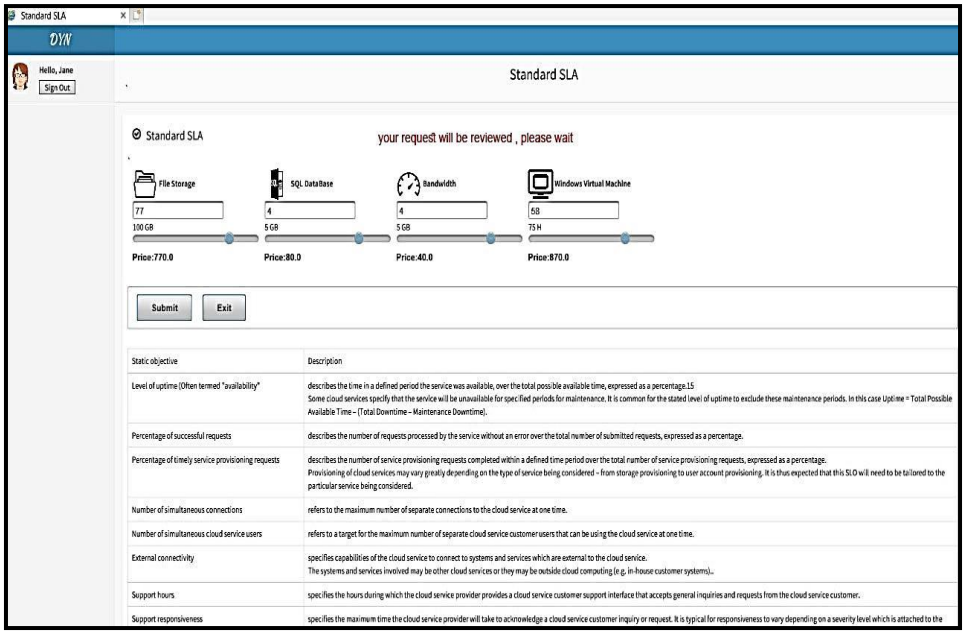

Fig (10) : user window for dynamic negotiation with third party, where the users will see a notification informing that their request is being reviewed and they will need to wait for a while.

The following window in Figure 11 illustrates an intelligent third party chooses whether to accept or reject a user's request.

Also, explaining in detail the user's set of changes, and the system requirements which represents the overall system capabilities from the desired resources .

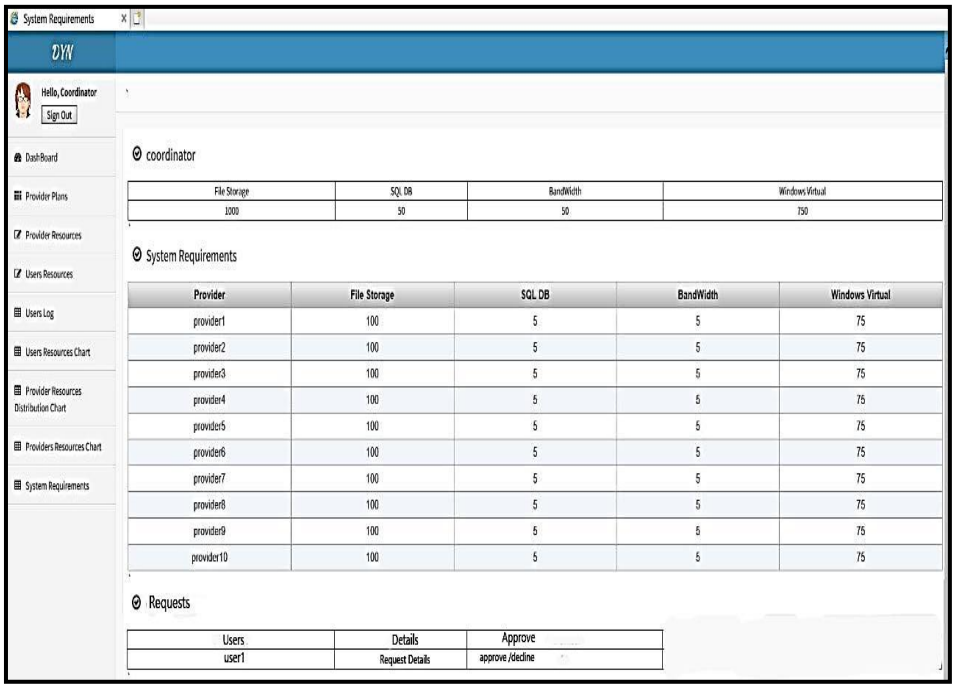

Fig (11): system requirements for each provider's and user's requests to be accepted or rejected during dynamic negotiation.

After approval for the change request, the user will be informed by mail, renegotiation is ready to be approved, he will be able to use the resources as showed in user dashboard page which displayed in Figure12.

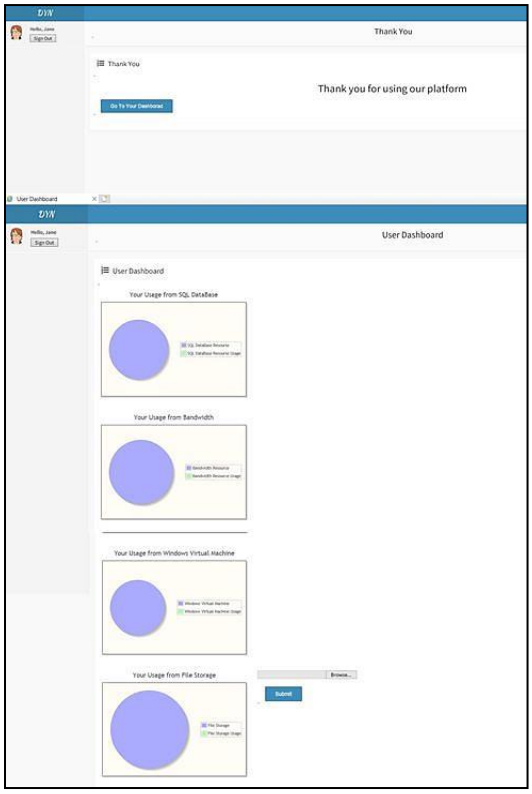

Fig (12): user dashboard، which shows the resources that have been reserved by the user and approved.

6. PERFORMANCE MEASUREMENT

The purpose of performance measurement is to track the effects of the dynamic agreement with third party on the various requirements of the reserved servers.

In order to properly measure performance, the following steps have been taken:

1- Reserving five instances servers with different specifications on Amazon Cloud as showed in Figure 13.

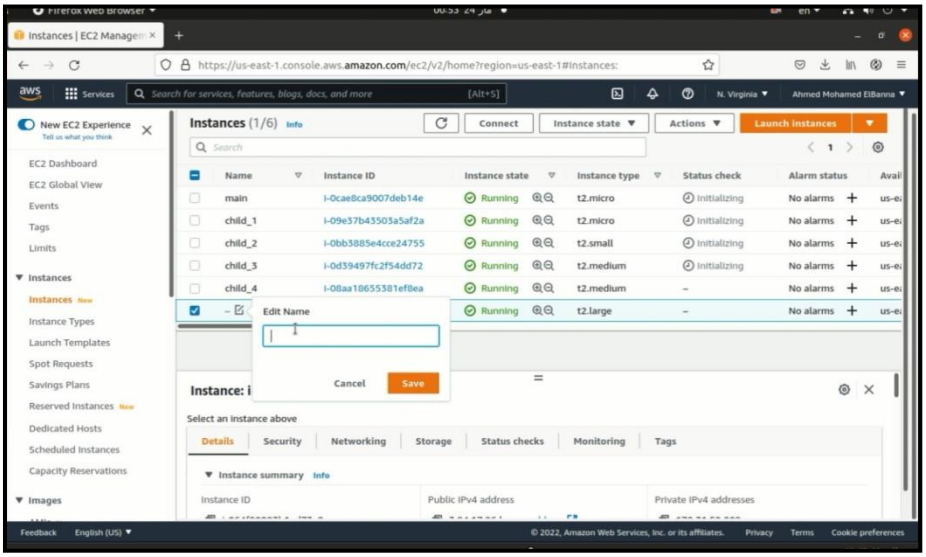

Fig (13): amazon cloud servers reservation with instances requirements, and parameters

2- Presenting a number of users to the proposed platform system.

3- Uploading the proposed platform's user database to the reserved servers.

4- Track modifications to servers parameters after loading users, Figure 14 declared the parameters modification .

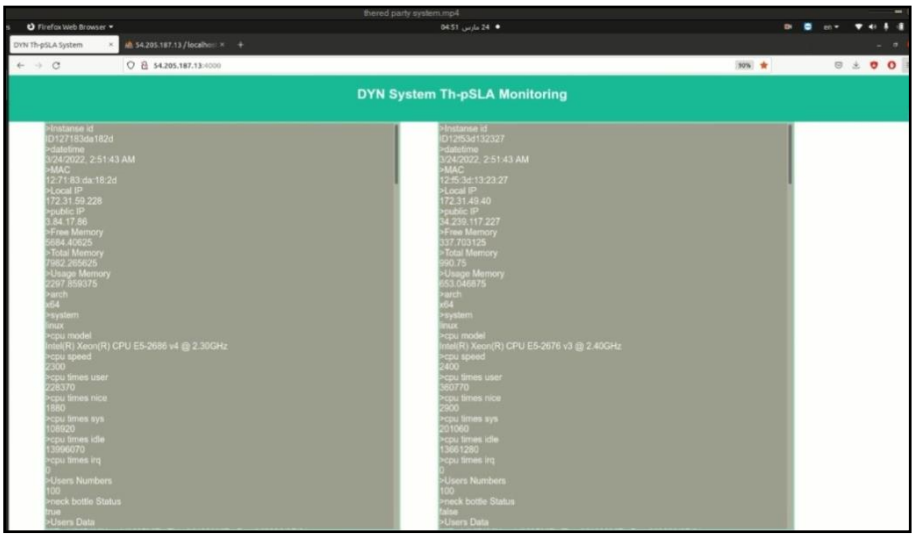

Fig (14): parameters modification for amazon cloud servers after uploading users.

Table 2. explains the users' influence on the various requirements of the reserved servers .

Instance specification	Initial Values					After Uploading Users				
	Instance 1	Instance 2	Instance 3	Instance 4	Instance 5	Instance 1	Instance 2	Instance 3	Instance 4	Instance 5
CPU times user/ clock ticks	18100	17180	42860	20520	3525	220490	22837	360770	227280	261210
CPU times sys/ clock ticks	10060	8970	16310	8840	14860	110730	108920	201060	100010	143710
CPU times idle/ clock ticks	473640	4840960	4423000	4727330	4438.160	14017010	13996070	13661280	14072080	13916030
Free memory	3551.8 Mb	7554.8 Mb	679.035 Mb	3556.2 Mb	1676.40 Mb	3597.85	5684.4	337.703	3591	1659.54
Usage memory	398.4 Mb	427.42 Mb	311.71 4 Mb	303.3 Mb	322.34 Mb	352.42	2297.8	653.046	358.8	339.2
Neck bottle	———	———	———	———	———	False	true	false	false	true

Theses specification considered as the most significant influencing aspects for study performance and monitor the proposed system behavior, it is observed that an increase in CPU clock ticks. Also, it is evident that a system throttle was present in instances 2 and 5 which affected the platform performance.

A comparison will be made in future work between this system and the complete dynamic system without the third party on the same requirements of the reserved servers.

7. Conclusions:

This paper's main contribution is the creation of a framework for dynamic service level negotiations for the cloud. This presented negotiating process framework primarily relies on intelligent agents that play the role of third parties to overcome obstacles that arose in fixed agreements, like the ongoing changes in business service requirements, improve dynamic negotiations, and keep up with the ever-changing character of the cloud environment, like multi-tenancy and various distributed systems.

8. Recommendations:

Using the automatic bargaining concept, Future work will involve designing and implementing a framework with multiple agents based on an automated environment that does remove the conventional of third party (broker) in an effort to overcome common weaknesses like security concerns. Future work will also involve performance evaluation and comparing between the two different schemes.

Conflict of interest: There is no conflict of interest.

9. REFERENCES

[1] D. Kumar , and T. Sharma" Issues and Challenges in Cloud Computing",International Journal of Advanced Research in Computer and Communication Engineering, Vol. 8, Issue 4, April 2019, ISSN 2278-121

[2] V. Paul, S. Pandita, and M.Randiva," Cloud computing review," International Research Journal of Engineering and Technology (IRJET), Vol.05, Issue: 03, pp. 1454-1456 , Mar.2018.

[3] M. Sajid , S. Sharma, "Integrated Fog and Cloud Computing: Issues and Challenges", International Journal of Cloud Applications and Computing ,vol11, no.4, February 2021 DOI:10.4018/IJCAC.2021100110.

[4] S. Samreen, N. Khatri-Valmik, S. Salve, and P.Khan ," research to cloud computing ," International Research Journal of Engineering and Technology (IRJET) , Vol.05 Issue: 02 , pp.785-788 , Feb.2018..

[5] D.El-mattary, F.Omara, S.Elhafeez, W.Awad, "SLA for E-Learning System Based on Cloud Computing", International Journal of Advanced Computer Science and Applications (IJACSA) Vol. 6, no 10, October 2015, doi:10.14569/IJACSA.2015.061027.

[6] E.Aljoumah, F.Al Mousawi, I.Ahmad, M. Al Shammri, and Z.AlJady, " SLA in Cloud Computing Architectures: A Comprehensive Study", international Journal of Grid Distribution Computing, Vol. 8, no.5,October 2015, doi:10.14257/ijgdc.2015.8.5.02.

[7] F.Zulkernine, and P. Martin , " An Adaptive and Intelligent SLA Negotiation System for Web Services" , IEEE TRANSACTIONS ON SERVICES COMPUTING, vol. 4, no. 1, April 2011, doi:10.1109/TSC.2010.44.

[8] M. Al-aaidroos, N. Jailani, and M. Mukhtar , " Agent-based Negotiation Framework for Web Service's SLA" , 7th International Conference on IT in Asia (CITA) , Jullay2011 , doi:10.1109/CITA.2011.5999537.

[9] Z.Xiao, D.Cao, C.You, and H.Mei, " A Policy-based Framework for Automated Service Level Agreement Negotiation" , IEEE International Conference on Web Services, July 2011, doi:10.1109/ICWS.2011.116.

[10] L.Wu, S.Garg, R. Buyya, C.Chen, and S.Versteeg, "Automated SLA Negotiation Framework for Cloud Computing" , 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, May 2013, doi:10.1109/CCGrid.2013.64.

[11] J.Chen, X.Han, and G.Jiang, "A Negotiation Model Based on Multi-agent System under Cloud Computing", ICCGI: The Ninth International Multi-Conference on Computing in the Global Information Technology, 2014 , ISBN: 978-1-61208-346-9 .

[12] A. Maarouf, A.Marzouk, and A.Haqiq, " Towards a Trusted third party based on Multi-agent systems for Automatic control of the quality of service contract in the Cloud Computing" , 1st International Conference on Electrical and Information Technologies ICEIT , vol.978, no.1,March2015, doi:10.1109/EITECH.2015.7162972.

[13] Z.Bakraouy, A.Baina , and M. Bellafkih ," System Multi Agents for Automatic Negotiation of SLA in Cloud Computing" , Springer International Publishing AG, part of Springer Nature, vol.735, March2018, doi:10.1007/978-3-319-76354-5 .

[14] F. Li, "A Trust Model for SLA Negotiation Candidates Selection in a Dynamic IoT Environment", IEEE Transactions on Services Computing,vol.15,no.5, April 2021, doi**:** 10.1109/TSC.2021.3070405.

[15] R.Rajavela, S.K.Ravichandranb, P.Nagappana ,and G. R. Kanagachidambaresan,"Cloud service negotiation framework for real-time E-commerce application using game theory decision system" , Journal of Intelligent & Fuzzy Systems ,vol.41 , no.5, April2021, doi:10.3233/JIFS-189882.

[16] Rafael Brundo Uriartea, Rocco De Nicolaa, Vincenzo Scocaa, Francesco Tiezzib .," Defining and Guaranteeing Dynamic Service Levels in Clouds", Journal of future generation computer systems, vol.99 , October 2019, doi.org/10.1016/j.future.2019.04.001.

[17] P.Srivastava, and R. Khan ," A Review Paper on Cloud Computing ," International Journals of Advanced Research in Computer Science and Software Engineering,Vol.8, Issue-6 , pp .17-20 , june. 2018, ISSN: 2277-128X