An Autonomous Multi-agent Framework using Quality of Service to Prevent Service Level Agreement Violations in Cloud Environment

Jaspal Singh*, Major Singh Goraya
Department of Computer Science and Engineering-Sant Longowal Institute of Engineering and Technology, Deemed University, Longowal, Sangrur, Punjab, India

Abstract—Cloud is a specialized computing technology accommodating several million users to provide seamless services via the internet. The extension of this reverenced technology is growing abruptly with the increase in the number of users. One of the major issues with the cloud is that it receives a huge volume of workloads requesting resources to complete their executions. While executing these workloads, the cloud suffers from the issue of service level agreement (SLA) violations which impacts the performance and reputation of the cloud. Therefore, there is a requirement for an effective design that supports faster and optimal execution of workloads without any violation of SLA. To fill this gap, this article proposes an automatic multi-agent framework that ensures the minimization of the SLA violation rate in workload execution. The proposed framework includes seven major agents such as user agent, system agent, negotiator agent, coordinator agent, monitoring agent, arbitrator agent and the history agent. All these agents work cooperatively to enable the effective execution of workloads irrespective of their dynamic nature. With effective execution of workloads, the proposed model also resulted in an advantage of minimized energy consumption in data centres. The inclusion of a history agent within the framework enabled the model to predict future requirements based on the records of resource utilization. The proposed model followed the Poisson distribution to generate random numbers that are further used for evaluation purposes. The simulations of the model proved that model is more reliable in reducing SLA violations compared to the existing works. The proposed method resulted in an average SLA violation rate of 55.71% for 1200 workloads and resulted in an average energy consumption of 47.84kWh for 1500 workloads.

Keywords—Cloud computing; multi-agent framework; SLA violations; energy consumption; history agent; Poisson distribution

I. INTRODUCTION

Cloud computing is a well-established paradigm that offers computing resources and services in a pay-as-you-go fashion to all the users connected to it [1]. It also provides resources to users that can be fully controlled by the users themselves through the virtualization of resources [2]. The cloud paradigm can be generally categorized as a scalable architecture that supports the inheritance of a wide range of technologies including utility computing, service-oriented architecture (SOA), and virtualization [3]. This paradigm also provides a shared pool of resources that offers services to workloads belonging to diverse forms of applications. Virtualized IT resources offer services of three types including software as a service (SaaS), Platform as a service (PaaS), and infrastructure as a service (IaaS) [4, 5]. With deep-spread data centers, the cloud paradigm ensures seamless services to its end users. Most of the popular organizations and companies are currently rendering cloud services to their customers and some of them include Google, Amazon, and Microsoft [6, 7]. The main acceptance of this paradigm is due to the flexible services offered where the users are requested to pay only for the services they have used [8].

The daily needs of the general community are satisfied with the cloud computing service which stays at a basic level of the computing paradigm [9]. Such a computing facility is specifically introduced to provide quality of service (QoS) aware services to a market of users to meet their objectives and requirements [10]. Thus, the service level agreement (SLA) oriented resource management is a crucial need for the users that negotiate a pile of virtualized and inter-connected systems between the users and cloud service providers or between the resource providers and brokers [11]. Due to the widespread availability of business models, it becomes a highly complex issue to select the appropriate service provider that can fulfil the execution of an application by meeting its QoS requirements [12]. A system-centric resource management framework is usually employed by cloud providers to offer computing services and resources [13]. A market-oriented resource management framework is of utmost need to enable the supply and demand of resources thereby offering feedback to both providers and consumers in terms of economic incentives [14]. Also, based on the usage of resources and services, the service requests are distinguished through QoS-based resource allocation [15].

Currently, the cloud paradigm provides only limited support for dynamic SLA negotiations between the associated participants such as cloud service providers and consumers [16]. Also, there are no reliable mechanisms that can offer automatic resource allocation to multiple competing requests [17]. The existing frameworks are unable to completely support customer-driven service management with the requested service requirements and customer profiles [18]. The SLAs that are signed between the cloud customers and cloud service providers are required to be maintained on each call of request processing and executions. Generally, market-based resource management strategies are more focused on customer satisfaction and service provider profits [19, 20]. Therefore, the development of a framework that can satisfy

both the service providers and customers is of utmost need [21]. In most of the research works conducted, it has been concluded that it is almost impossible to extract appropriate market-based resource management schemes that can encompass both computational risk management and user-driven service management to sustain the SLA-aware allocation of resources [22, 23].

The SLA-oriented schemes are required to offer personalized attention to customers to help them meet their SLA-aware objectives [24]. One of the most important factors to be considered while designing such a solution is that the demands of the users fluctuate with time for the changes encountered in the operating environment and business operations [25]. SLA can be defined as a formal agreement that provides information regarding the quality of every non-functional requirement (NFR) of a service [26]. A formal procedure is followed in cloud computing that if there is any SLA violation encountered in the workload execution process, then penalties are provided to the service providers [27]. When there are no violations of SLA for different workload executions, then rewards are provided either to the customers or service providers after evaluations [28]. One of the major problems arising here is with the dynamic execution of workloads where there are a huge number of workloads arriving in the cloud for executions. At this point, the QoS cannot be assured in every circumstance and there is a requirement for an automated system that can accurately monitor the violations occurring within the environment [29, 30]. Therefore, there is a leading requirement for an automatic system that can control and monitor the QoS of the workloads within the negotiated terms.

A. Motivation

There are several techniques encountered to automate the process of resource management via SLA negotiation. Generally, those methodologies integrate virtualization and market-based allocation policies for allocating the cloud resources to workloads to complete executions. Several efforts have been made to automate the process of SLA-aware resource allocation to the workloads. Some methods focused on framing SLA to workload execution through the negotiation process whereas others focused on automating the entire process. But, only a few methods explored the benefits of multiple agents in the cloud to enable the automatic management of resources to support SLA-aware workload execution. Therefore, there is a need for such a technique to be enforced to avoid SLA violations while executing the workloads. Therefore, this paper presents an automatic multi-agent framework that supports the execution of workloads without any violation of SLA. Moreover, the proposed framework also optimizes energy consumption in data centers to enhance overall performance.

B. Contribution

The major contributions of the proposed work include the following:

- A new and efficient multi-agent system is proposed in this work to enable seamless services to its users by satisfying their fluctuating demands and enabling SLA-aware executions of workloads.
- Presenting the agent-based cloud framework where each of the agents is incorporated to provide timely execution of workloads without disturbing the SLAs. Moreover, the framework is designed in a unique way to satisfy both the service providers and the customers involved.
- Introducing an additional history agent within the agent-based framework to keep track of the resources used and the requests processed. The aim of adding this agent is to enable the prediction of future demands so that the overall efficiency and reputation of the system can be enhanced.
- Introducing the Poisson distribution function (PDF) model to generate random numbers based on the input to form the dataset. The generated dataset is then provided to the proposed model to evaluate and compare the model extensively.

C. Organization

The remainder of the paper is structured as per the following: Section II presents the literary works established by other researchers working in the same field, Section III provides the proposed methodology with architectures and explanations, Section IV provides the results and discussion with comparative analysis and Section V concludes the paper with future scopes.

II. Background on Resource Provisioning in Cloud Computing and QoS Constraints

Some of the recent works established for controlling SLA violations in the cloud are reviewed below:

Cloud computing technology faces several challenges among which SLA violation is one of the most common and tiring problems affecting its overall performance. In the cloud-based e-commerce negotiation framework, the optimization of broker negotiation strategy is a cumbersome task. Generally, long-term or pre-request optimizations are followed to resolve the task. The pre-request strategies focus on the usage of various utility functions and are followed in most research works. The long-term strategies are less focused and most of them are unable to guarantee negotiation and state-of-art to minimize SLA. Such limitation was addressed by Rajavel and Thangarathanam [31] effectively through the stochastic behavioral learning negotiation (SBLN) technique. The main intention of the technique was to maximize the success rate and utility value to a maximum level. The increase in the desired values was attained by increasing the count of negotiation rounds. The performance of the method was implemented and compared with other techniques and the outcomes proved its efficacy.

The Multiple agent-based systems were developed by Azhagu and Gnanasekar [32] to deal with the SLA violations in the cloud computing infrastructure. Violations of SLA affect the business operations of both the cloud service providers and customers as compensation is required to be provided by the service providers (CSP) for their customers. The agent-based model enhanced the trust of every stakeholder through the automatic minimization of SLA

violations. The framework included a total of six agents a user agent, a system agent, a negotiation agent, a coordinating agent, a monitoring agent, and an arbitrator agent. The monitoring agent was responsible to monitor the cloud environment and indicated SLA violations. The arbitrator agent observed and identified the cause of the violation and posted penalties or rewards based on the performance. After evaluations of the entire framework, the outcomes suggested that the method was effective in controlling SLA violations in the cloud with the maximization of performance in workload executions.

The Discovery of cloud services is a highly challenging issue due to the increase in complexities and network size. With the dynamic increase of these two factors, the effective discovery of services is hampered making it an NP-hard problem. The popular cloud service discovery method based on ant colony optimization (ACO) suffered from load balancing issues. To resolve the issue and enable effective usage of resources, Heidari and Navimipour [33] introduced the inverted ACO (IACO) method that promised load-aware service discovery to the cloud. In the inverted algorithm, the attractive behavior of pheromones was replaced with the repulsive behavior. The model was simulated using the Cloudsim tool and the numerical results of the model proved its efficiency over the other compared methods. Also, the model provided several other benefits including energy efficiency, response time mitigation, and control of SLA violations.

Cloud computing supports large-scale processing in a distributed fashion with higher flexibility. SLA violations in the cloud occur due to several facts and it is important to control these violations to attain performance improvement. VM allocation is one of the common and challenging problems in the cloud resulting in SLA violations. Other problems associated with VM allocation include problems in asset utilization and energy consumption. An SLA-aware strategy to allocate the VMs in the cloud using an intelligent algorithm was introduced by Samriya et al. [34]. To attain the objective, the method utilized the multi-objective emperor penguin optimization (EPO) algorithm that allocated the VMs in a heterogeneous cloud environment. Further, simulations were conducted to prove the performance of the method compared with other multi-objective metaheuristic optimization algorithms. The outcomes proved that the model effectively reduced SLA violations and energy consumption in the cloud environment.

Another strategy based on resource allocation was introduced by Belgacem et al. [35] based on the exploration of properties of multiple agents in the cloud. Cloud infrastructure face challenges in resource allocation due to its heterogeneous nature, volatile resource usage, and accommodation of VMs with diverse specifications. The method introduced the combination of an intelligent multi-agent system with the reinforcement learning method (IMARM) to attain the objective of optimal resource allocation. The Q-learning process was combined with the properties of multiple agents to gain performance enhancement in resource allocation accordingly. IMARM method responded well to the fluctuating customer demands through dynamic allocation and release of resources accordingly. Moreover, the VMs were moved to the best state concerning the current state environment through the learning model. Finally, simulations were conducted to prove the performance improvement attained by the model compared to previous models in terms of various metrics.

In both cloud and utility-based computing platforms, SLA emerge as a chief aspect while providing personalized services to the users. In order to offer flexible establishment of SLAs and to prevent SLA violations, Son and Jun [36] presented a proactive resource allocation (PRA) scheme. The presented scheme optimally selected a suitable datacenter among the available globally distributed datacenters to enhance resource allocation to the workloads. The method also provided time slots and price negotiations for flexible SLAs. The effectiveness of the method was proved through experiments.

TABLE I. COMPARATIVE ANALYSIS OF THE EXISTING LITERARY WORKS

Authors	Methods	Advantages	Drawbacks
Rajavel and Thangarathanam [31]	SBLN	Obtained drastic increase in success rate and utility value	The unwanted conflicts among the participants are required to be addressed
Azhagu and Gnanasekar [32]	Multi-agent system	Automatically controlled SLA violations in the cloud through continuous monitoring	More QoS parameters are required to be considered to attain optimal performance
Heidari and Navimipour [33]	IACO	Efficient discovery of cloud services with enhanced utilization of resources	The repulsive behavior of pheromones is required to be evaluated deeper to prove its advantages
Samaria et al. [34]	Multi-objective EPO	To enable effective VM allocation in the cloud with minimized SLA violation and energy consumption	Other important performance objectives such as wastage of resources are required to be focused
Belgacem et al. [35]	IMARM	Dynamic allocation and release of resources and providing better responses to the customers' changing demands	More metrics are required to be considered and more analyses are required to prove its reliability
Son and Jun [36]	PRA	The overall efficiency of negotiation and utility has been increased with the trade-off algorithm	Limited SLA options are provided by the framework based on enforced SLA strategies
Wu et al. [37]	PURS	Facilitated intelligent bilateral bargaining of SLAs and provided maximum profit for the brokers through enhanced customer satisfaction levels	The penalty for the failure of negotiation from the user's perspective is not considered

Another SLA negotiation framework was introduced by Wu et al. [37] to accomplish profit with higher customer satisfaction. The process of negotiation establishment becomes tough with the existence of multiple CSPs. The introduced framework considered SaaS broker as a one-stop-shop for the customers and negotiation was performed with multiple CSPs. The automated framework supported bilateral bargaining of SLAs and helped in maximizing the profit of brokers. Extensive evaluations with real CSP proved the efficacy of the method. Table I presents a comparative analysis of the existing literary works.

A. Problem Statement

On reviewing the existing works, it has been identified that the multi-agent system in the cloud is highly advantageous and helps to offer numerous reliable services to its customers. The Multi-agent-based framework is one of the effective methods to enable the execution of workloads without any violation of the SLA constraints. The existing methodologies are unable to completely enable the execution of workloads within the defined deadlines. Other agent-based frameworks are merely unstable as the failure of negotiation is not given importance or considered that may result in performance degradation. Moreover, the negotiations terms and conditions are not well-established in most of the existing works. Apart from these, the demands of the future workloads are unidentified which delays processing of workloads. Because of looking forward to attaining optimal resource provisioning using QoS in cloud computing and higher performance by satisfying the QoS constraints of users, a very few techniques are formulated based on the self-management of cloud services using multiple agents. Moreover, it is of utmost need to optimize the violations of SLA with the help of negotiation before the deployment of services in the cloud. To overcome the existing drawbacks and to fill the gaps, a new multi-agent-based framework is introduced based on the accommodation of multiple agents to monitor and complete the execution of workloads within the defined SLA. The proposed framework also utilizes an additional agent to back up the details regarding executions in order to identify the future demands for resources. By this way, the profit and rewards from both the ends can be considered and the effectiveness of negotiations can also be improved.

III. Proposed Methodology

Execution of cloud workloads within the defined deadlines is a complex task and requires appropriate algorithms and techniques. Efficient workload execution in cloud is highly crucial as it has wide range of applications supporting companies associated with it. The agent-based frameworks are faster in approaching the requests from users compared to other agentless frameworks. This helps to complete the workload execution within the deadlines. Moreover, these frameworks are capable of constantly monitoring the environment and collect data at real time. Due to these advantages, a new Autonomous Multi-agent-based framework based upon Probability and History (AMAPH) is designed in this work to prevent SLA violations and to attain higher performance in workload executions. The proposed multi-agent system monitors the cloud environment and checks for SLA violations. When there is no violation encountered in a workload execution, rewards are provided to the service provider or customer and when there is a violation, penalties are provided and the reason for the violation is determined. The proposed mechanism works deliberately to avoid any kind of SLA violation within the cloud environment and assures proper execution of workloads that are succeeded respectfully. Moreover, the History agent keeps a record of either successful or failed requests, respectfully. The overall architecture of the proposed work is displayed in Fig. 1.

The proposed multi-agent framework includes seven agents a user agent, a system agent, a negotiation agent, a history agent, a coordinating agent, a monitoring agent, and an arbitrator agent. The requests reach the user agent at the initial stage and then based on the type of request; it is forwarded to the system agent.

The type of service required for processing the request is determined and the request is forwarded to the negotiation agent where a negotiation process is initiated between the user agent and service provider. A service is selected for the request and the details are then forwarded to the coordinating agent. The history agent is responsible to track the services offered to the requests. SLA is established by the coordinator and the monitoring agent dynamically monitors the environment for any violation and each violation, an indication is sent to the arbitrator agent.

Finally, penalties are laid by the arbitrator to the service provider and the type and reasons for the violation are determined. The Poisson distribution function (PDF) component is included in the framework to test the performance of the proposed system (AMAPH) and assuring for the different number of workloads accordingly in comparison to the base paper [32], at a glance.

A. User Agent

The user agent is the initial agent of the proposed framework, and the role of the agent is to receive the requests provided by the associated cloud users. Thus, the cloud users directly request the services via the user agent to the cloud. This agent is responsible for dealing with the user registration processes for new users. Each user is linked with a single user agent to attain the cloud services.

For any kind of additional services requested by the user, multiple user agents are not created in the proposed work and the additional requested services are handled by the same agent. For registration of new users, the user agent gathers the required information such as the personal information of users via a registration form. In the case of service requests from the user side, the user agent determines the type of service being requested by the user. All the details regarding the service type requested are collected and analyzed and then the requests are forwarded to the system agent for further processing.

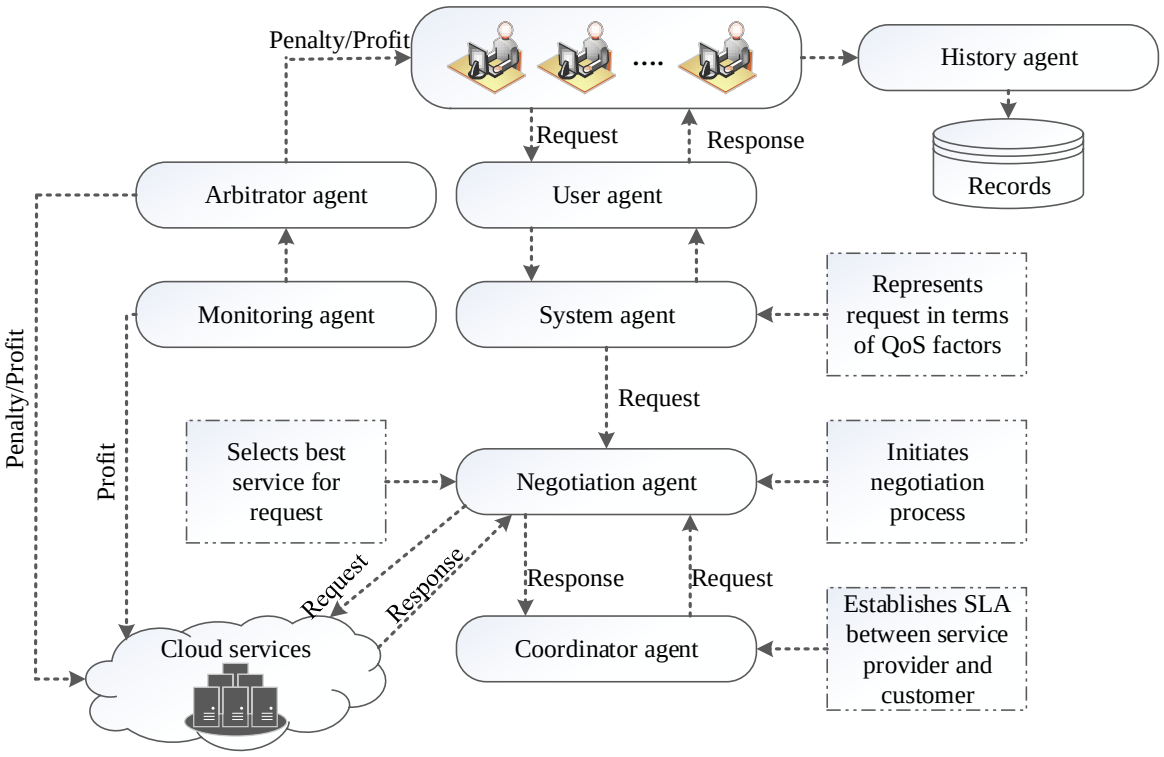

Fig. 1. Architecture of the proposed Multi-Agent framework (AMAPH).

B. System Agent

The system agent receives the requests from the user agent and determines the actions to be taken further. The details regarding the requests are obtained and then the requests are represented in technical terms including the quality of service (QoS) factors such as account type, number of accounts, contract length, solution time, and response time [32]. The system agent is responsible for verifying the quality factors of all the incoming requests and helps the framework to better process the requests. After representing the requests in terms of quality factors, the service type requested by the user is identified. Based on the requested service type, the system agent either forward it to the negotiation or the coordinator agent.

C. Negotiation Agent

This agent is responsible for initiating the negotiation process between the user agent and the service provider. The negotiation process is established based on diverse technical factors including nature of service, reliability, response time, monitoring, reporting of service, and responsibilities. Based on the technical factors and the service type being requested by the user, the negotiation agent communicates with the available service providers. The available service providers on the other side, place bids in the given view of processing the requests based on the available resources, resource capabilities, market circumstances, and business objectives. The main significance of the proposed framework is that the negotiation agent broadcasts the request details to all the service providers to provide the best service to the requests. The negotiation process ensures maintaining a more feasible SLA in workload executions. Based on the requested details available, the negotiation agent evaluates the received bids from service providers. Then, the attributes of service providers are compared with the resource requirements of the user and the appropriate service provider has selected that best suit the request. The details regarding the selected service and the service provider are then forwarded to the coordinator for further processing. Further, the history agent shall maintain the state-of-art in records wherein the id of the user, the CSP being selected by the negotiator as optimal resource provisioning process using QoS with respect to different data centres.

D. Coordinator Agent

The coordinator agent receives the request and selected service details from the negotiation agent and evaluates the request. The agents evaluate the received request for first-time access or request for service upgradation. After analyzing the type of request received, appropriate actions are taken further. The agent also formally establishes an SLA between the respective user and service provider and the message is forwarded to both parties. Apart from sending the message, it is also preserved by the agent for enforcement. Finally, the SLA is sent to the monitoring agent for further effective actions.

E. Monitoring Agent

The main responsibility of the monitoring agent is to continuously monitor for SLA violations within the cloud environment. Based on the established SLA details received from the coordinating agent, the monitoring process is regulated by the agent. When a violation is encountered in the environment, the agent immediately sends an indication to the arbitrator to take appropriate actions or to provide a penalty to

the respective party. If there is no violation in the workload execution, then the monitoring agent sends an indication about providing a profit message to the respective party for the successful execution of the task. It recommends the arbitrator provide rewards to the concerned service provider.

F. Arbitrator Agent

This agent is responsible to analyze the type of violation that has occurred and the reasons behind the occurrence of such violation. Then, based on the analysis, penalties are enforced on the service providers or the respective customers concerning the defined SLAs.

G. History Agent

The history agent is one of the significant agents in the proposed work that keeps track of service usage and workload executions. This helps the system to predict future workload requests and the type of services that could be predicted by those requests. The history agent maintains records where the id of the user, the type of service requested, the service being selected by the negotiator as optimal resource provisioning, and the service provider allocated to process the requests are stored as files. Based on these details, the agent predicts future workload requests and the type of service needed to process the request. By predicting these parameters, the proposed system decides on faster workload executions with minimized SLA violations. The history agent keeps a record wherein the service provider allocated to the request and other constraints are stored as files.

H. Poisson Distribution for Random Number Generation

The Poisson distribution function (PDF) is followed in this work for random number generation and these numbers are then given to the model for evaluation purposes. This distribution has very minimum parameters and is very simple to implement. Therefore, this distribution is chosen in our work to reduce the complexities. Consider a discrete random variable χ and it is assumed to follow a Poisson distribution with parameter $\lambda > 0$ if and only if it follows the following probability mass function:

$$f(k;\lambda) = \Pr(\chi = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (1)$$

where, k specifies the count of occurrences, e is the Euler number, λ specifies the positive real number which is equal to the expected value and variance of the random variable χ.

$$\lambda = \mathrm{E}(\chi) = \mathrm{var}(\chi) \quad (2)$$

This distribution can be generally followed in systems with a large number of rare but possible events and the count of such events within a fixed time interval can be specified as a random number with the Poisson distribution. Instead of knowing its value of λ it the system provides the value of the average rate δ, then λ is substituted by δt an Eq. (1) that can be adopted as follows:

$$\mathrm{P}(k \text{ events within int } t) = \frac{(\delta t)^k e^{-\delta t}}{k!} \quad (3)$$

IV. Results and Discussion

A detailed analysis of the results obtained through evaluations of the proposed framework is presented in this section. The entire simulations of the proposed work have been carried out using the CloudSim tool with the Java Agent Development Environment (JADE). The proposed system includes multiple active agents such as a user agent, system agent, negotiation agent, history agent, coordinator agent, monitoring agent, and arbitrator agent. The user agents receive the requests and provide a registration form if it is a new request or evaluates and forwards the details to the system agent. The system agent evaluates the requests and represents them in technical form and then forwards it to the negotiation agent where the negotiation process is initiated. The history agent is responsible to keep track of certain important records and the coordinator agent chooses the appropriate service to process the request based on the SLAs. The monitoring agent monitors the entire cloud environment for SLA violations and if there is any violation, then the agent sends an indication to the arbitrator for penalty enforcement or directly forwards a message for rewards if there is no violation. The proposed framework is autonomic and automatically monitors and controls the environment without the need for the intervention of a cloud engineer.

In the JADE environment and evaluations, the overall framework is implemented as agents and the random numbers are generated via the Probability Distribution Function for realistic datasets that are exchanged between agents for collaboration. Apart from the seven agents of the framework, the environment also accommodated a resource manager, cloud broker, VM manager, physical machine manager, and cloud registry. The entire simulations are carried out with a total of 05 data centers and 20 service providers. In the simulations, the requests arising from the VMs are forwarded to the service broker. The workloads are simulated based on the business workload traces provided by GWA-T-12 Bitbrains. The simulated dataset includes the performance values for different VMs running in datacenters and the data are recorded in .CSV files. The data values are generated for 5 datacenters to provide extensive evaluations and analysis. The generated dataset included performance values such as CPU usage, memory usage, network throughput, disk throughput, CPU capacity provisioned and memory capacity provisioned. A total of 1500 workloads are generated to evaluate the proposed system and each workload included the above-mentioned performance values. Moreover, the dataset consisted of no missing or duplicate values and this reduced the need for preprocessing. For comparison, the proposed method selected PRA [36], PURS [37], and a multi-agent system [32]. All these results are respectfully taken from the multi-agent system [32] for comparison; the results are undertaken by varying the number of workloads, resources, and execution times for optimal resource provisioning using QoS in cloud computing.

A. Performance Metrics

The proposed framework has been evaluated in terms of SLA violation rate and energy consumption [32]. The mathematical representations and descriptions of the metrics chosen are as follows:

SLA violation rate: SLA violation rate indicates the rate of violations occurring in the environment for different workload executions. The mathematical representation for the SLA violation rate can be given as follows:

$$S_{VR} = f_R * SLA_W \quad (4)$$

where, f_R is the failure rate and SLA_W is the weight of SLA. The failure rate can be measured using the following formulation:

$$f_R = \frac{W_{fR}}{W_{total}} \quad (5)$$

wherein, W_{fR} is the workloads' failure rate and W_{total} indicates the total count of workloads involved. The SLA violation rate is taken by varying the number of workloads, number of resources, and the number of execution times which is given by the following formulation:

$$E_T = \frac{W_{ct} - W_{st}}{W_{total}} \quad (6)$$

where, W_{ct} is the completion time of workload, and W_{st} is the submission time of workload.

Energy consumption: Energy consumption indicates the consumption of energy by the VM to complete the execution of a workload. The mathematical formulation is as follows:

$$E_C = (l * \max) + (1 - l) * \max E_{VM} \quad (7)$$

wherein, E_{VM} indicates the energy consumed by VM and l indicates the constant set to 0.5 in simulations.

B. Performance Analysis

The overall performance of the proposed framework is analyzed in this section. The simulations are performed with user-defined QoS constraints like CPU, RAM etc. All the results obtained are compared with the methods such as PRA [36], PURS [37], and multi-agent system [32]. The existing methodologies also follow the same configurations and parameter settings. The analysis of the obtained results is presented:

The results of the SLA violation rate for the different workloads are recorded. The performance of the proposed method is more optimal than the other methods. The addition of a history agent helped the model to accurately predict future workloads so that the SLA violation rates are reduced. The results are taken by varying the number of workloads from 0 to 1200. For all the workload input, the proposed model maintained higher performance compared to the other models. When the number of workloads is low, the violation rate is also low and when the workload is increased, the violation rate is scanty also and gradually increased. The performance comparison of SLA violation rate with respect to number of workloads is presented in Table II. A graphical representation of the results is presented in Fig. 2. The figure shows that there is only a minimal increase in violation rate for the proposed method showing its efficacy. The graph also shows that there is a huge impact on the overall performance of the framework when the number of workloads are varied. The proposed approach depicts a result of 19.5% violation rate when the number of workloads is 200 and resulted in 55.71% violation rate when the number of workloads is increased to 1200. Among the compared methods, the multi-agent system resulted in better performance compared to PRA and PURS and other details of QoS [32].

TABLE II. PERFORMANCE VALUES OF SLA VIOLATION RATE VS. NUMBER OF WORKLOADS

Methods	Workloads					
	200	400	600	800	1000	1200
PRA	41.81	46.88	47.11	57.99	69.06	79.35
PURS	32.52	37.2	42.46	50.44	62.28	69.48
Multi-agent system	23.35	28.3	36.64	45.38	56.52	59.47
Proposed	**19.5**	**26.79**	**32.03**	**38.41**	**46.21**	**55.71**

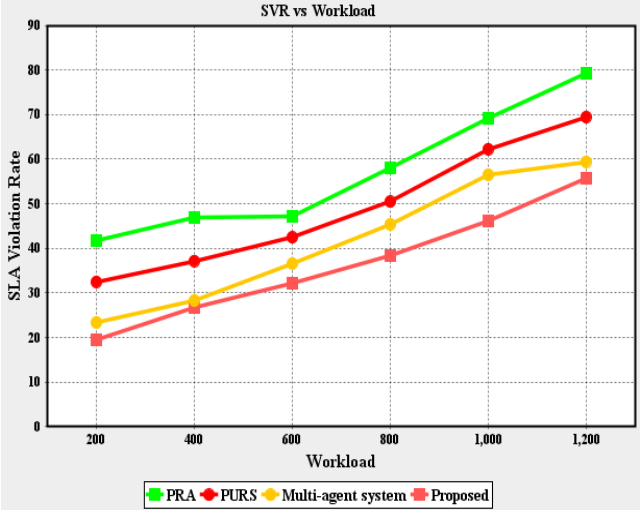

Fig. 2. Graphical representation of SLA violation rate vs number of workloads.

TABLE III. PERFORMANCE VALUES OF SLA VIOLATION RATE VS. NUMBER OF RESOURCES

Methods	Resources					
	50	100	150	200	250	300
PRA	90.91	166.51	195.22	235.41	310.05	354.07
PURS	81.34	145.45	157.89	220.1	282.3	309.09
Multi-agent system	64.72	113.08	140.76	197.22	269.85	290.35
Proposed	**48.29**	**91.27**	**110.91**	**134.88**	**164.11**	**199.77**

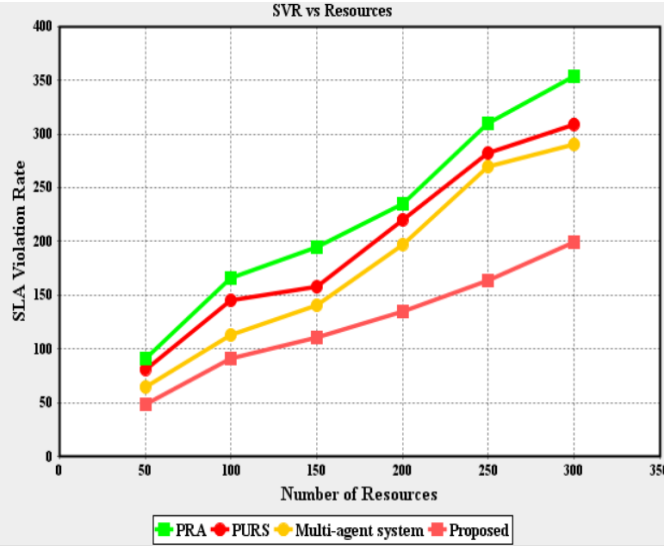

Fig. 3. Graphical representation of SLA violation rate vs number of resources.

The results of the SLA violation rate based on the number of resources are presented. The proposed technique is more optimal in resource provisioning using QoS than the existing methods in reducing SLA violations accordingly. The number of resources used for executing the workloads has a major impact on the variations in SLA violations. When the number of resources used in execution is less, the SLA violation rate is low significantly and when the resource is increased, the violation rate is also increased gradually in a scanty manner. This is because of the increase in the number of resources required to handle more workloads that results in increased SLA violations. This is also plotted in the graphical representation shown in Fig. 3. The values obtained on comparison of SLA violation rate with respect to number of resources are shown in Table III. The proposed method resulted in 48.29% of violations whereas for a total of 300 resources of violation rate could be affirmed accordingly. Among the compared techniques, the multi-agent system yielded better results.

The results of energy consumption for the different workloads are presented. The proposed method consumed less energy as compared to other methods in workload execution. When there is a minimum number of workloads, the energy consumption is less, and it increases gradually with the scanty increase in the number of workloads as per the provisioning of resources using QoS. This is also shown in the graphical representation presented in Fig. 4. The values obtained for energy consumption comparison are presented in Table IV. For 250 workloads, the energy consumed by the proposed method is 17.67kWh and for 1500 workloads, the energy consumed is 47.84kWh. Among the compared techniques, the multi-agent system consumed less energy to execute the workloads, and the other two methods consumed more energy for executions.

The results of the SLA violation rate for different execution times are recorded. The proposed method is more optimal than the existing methods. The values obtained on comparison of SLA violation rate with respect to execution time are shown in Table V. The graphical representation of the SLA violation rate based on execution times is shown in Fig. 5. The figure shows that the SLA violation rate is low for smaller execution times and gradually increases with the increase in execution times. Also, the proposed multi-agent system (AMAPH) produced better results as being compared to existing methods [32].

TABLE IV. PERFORMANCE VALUES OF ENERGY CONSUMPTION VS. NUMBER OF WORKLOADS

Methods	Workloads					
	250	500	750	1000	1250	1500
PRA	35.87	41.12	52.72	54.79	73.8	72.7
PURS	26.18	36.13	48.11	47.74	63.78	64.52
Multi-agent system	20.63	26.9	38.03	42.94	56.21	59.17
Proposed	**17.67**	**23.38**	**27.87**	**33.31**	**39.89**	**47.84**

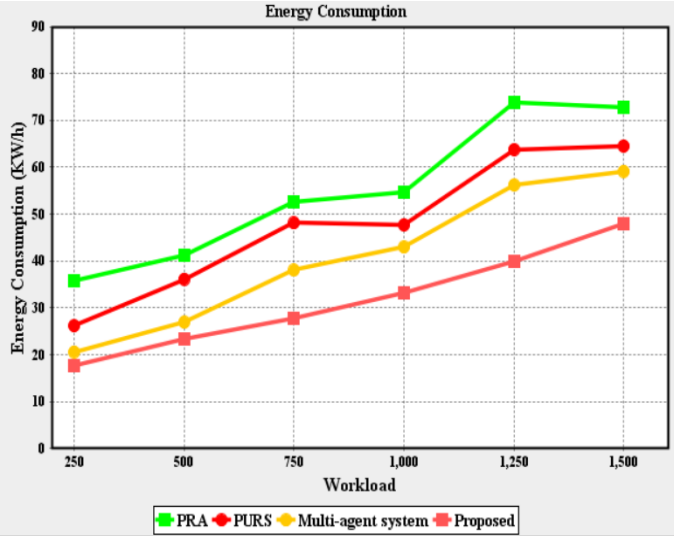

Fig. 4. Graphical representation of energy consumption vs number of workloads.

TABLE V. PERFORMANCE VALUES OF SLA VIOLATION RATE VS. EXECUTION TIME

Methods	Execution time						
	10	20	30	40	50	60	70
PRA	235.29	378.52	421.99	508.95	718.67	780.05	813.3
PURS	212.28	273.66	327.37	473.15	682.26	726.34	780.05
Multi-agent system	179.1	236.32	293.53	420.4	589.55	701.49	694.03
Proposed	**93**	**114.05**	**138.28**	**176.84**	**218.99**	**300.92**	**363.3**

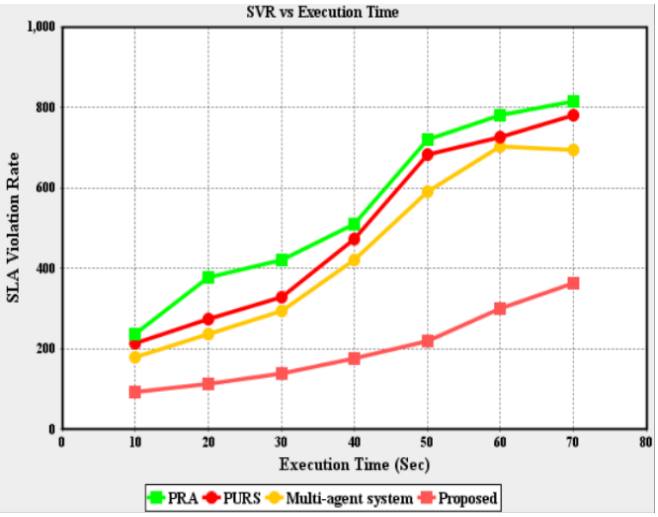

Fig. 5. Graphical representation of SLA violation rate vs execution time.

The overall simulations suggested that the proposed framework of resource provisioning using QoS is more optimal than the other compared techniques in reducing the SLA violations occurring in the cloud environment. As per the above results, the simulation build time has been considered accordingly but it may have a little change as in ground level of implications. The violation rate generally increases when the number of executions is increased. With the increase in the number of workload executions, the performance of the cloud network slows down due to higher energy consumption and increased violation rates as different case studies analyses for 05 different data centers. Therefore, the proposed framework is presented that is highly optimized for reducing the SLA violation rate and energy consumption in data centers. The performance of the framework is analyzed by varying the number of resources and workloads as these are the major factors influencing the performance. The results also proved that SLA violations increase when the number of executions needed is increased. While dealing with more workloads, the elasticity of the cloud and the resource availability is required to be regularly maintained to reduce SLA violations. The proposed framework of modelling and simulation measures the Quality of Service (QoS) and performance in Data-Center along with resource utilization policy. The analysis proved that the proposed model worked on reducing both the energy consumption in data centers and SLA violations in every dimension. The inclusion of a history agent within the architecture helped the model to forecast the arriving workloads and to predict the future requirement of resources. It kept track of the records of utilized resources and the available resources to maintain normal execution without any deviation in SLAs. The conjecture can be clarified for the response time as one of the major components of QoS factors based upon different workload's build time (Minimum is 21 sec and Maximum is 113 seconds). The method optimized the workload executions thereby reducing the overall violation rates and enhancing the overall cloud performance. Therefore, the proposed framework can be suggested as a promising tool to mitigate SLA violations and issues of higher energy consumption in cloud data centers and to achieve optimal performance for QoS as MOHFO and CGR analyses [15].

V. Conclusion

Cloud computing technology is one of the most popular computing technologies followed by most organizations throughout the world. This is because of its elastic and distributed nature that is capable of supporting faster network services with abundant provisioning of resources. In this work, a new and efficient framework is designed that supports the optimal execution of workloads with minimized SLA violations and energy consumption. The proposed framework includes multiple agents such as a user agent, system agent, negotiation agent, history agent, coordinator agent, monitoring agent, and arbitrator agent. The user agent obtains the request details from users and forwards them to the system agent where the technical terms of the requests are explored. The negotiation agent initiates the negotiation process between the service provider and the customer to avoid SLA violations. It selects the best service that can execute the current workload without SLA violation and with minimum consumption of energy. The history agent keeps track of workload executions to provide better forecasts of future executions. The coordinator agent receives the selected service details from the negotiator and establishes a formal SLA. The monitoring agent monitors the environment continuously for violations and sends an indication to the arbitrator if any violation is encountered. The arbitrator provides penalties or rewards to the service provider or customer and analyses the cause of the violation. The method is simulated and evaluated using a random number generated by Poisson distribution. The analysis proved that the method minimized the SLA violation rate and energy consumption in data centers much better compared to other existing techniques. Therefore, a resource provisioning framework using QoS attribute requirements to manage the resources of the Cloud while taking into account the Customer's Quality of Service as determined by the Service-Level Agreement (SLA) in the Cloud Computing environment has been incorporated successfully.

VI. Conflicts of Interest

The author declares no conflict of interest.

Author's Contribution

Conceptualization, Methodology, Software, Analysis, Resources & Investigations: Jaspal Singh. Supervision: Dr. Major Singh Goraya (my respected guide to PhD).

References

[1] K.S.S. Kumar, and N. Jaisankar, "An automated resource management framework for minimizing SLA violations and negotiation in collaborative cloud." International Journal of Cognitive Computing in Engineering vol. 1, pp. 27-35, 2020.

[2] S. Tuli, S.S. Gill, M. Xu, P. Garraghan, R. Bahsoon, S. Dustdar, R. Sakellariou et al., "HUNTER: AI based holistic resource management for sustainable cloud computing." Journal of Systems and Software vol. 184, pp. 111124, 2022.

[3] S.S. Gill, I. Chana, M. Singh, and R. Buyya, "RADAR: Self-configuring and self-healing in resource management for enhancing quality of cloud services." Concurrency and Computation: Practice and Experience vol. 31, no. 1, pp. e4834, 2019.

[4] M.A. Haghighi, M. Maeen, and M. Haghparast, "An energy-efficient dynamic resource management approach based on clustering and meta-heuristic algorithms in cloud computing IaaS platforms." Wireless Personal Communications vol. 104, no. 4, pp. 1367-1391, 2019.

[5] Y. Jararweh, M.B. Issa, M. Daraghmeh, M. Al-Ayyoub, and M.A. Alsmirat, "Energy efficient dynamic resource management in cloud computing based on logistic regression model and median absolute deviation." Sustainable Computing: Informatics and Systems vol. 19, pp. 262-274, 2018.

[6] D. Saxena, A.K. Singh, and R. Buyya, "OP-MLB: An online VM prediction based multi-objective load balancing framework for resource management at cloud datacenter." IEEE Transactions on Cloud Computing 2021.

[7] S.S. Gill, S. Tuli, A.N. Toosi, F. Cuadrado, P. Garraghan, R. Bahsoon, H. Lutfiyya et al., "ThermoSim: Deep learning based framework for modeling and simulation of thermal-aware resource management for cloud computing environments." Journal of Systems and Software vol. 166, pp. 110596, 2020.

[8] N. Gholipour, E. Arianyan, and R. Buyya, "A novel energy-aware resource management technique using joint VM and container consolidation approach for green computing in cloud data centers." Simulation Modelling Practice and Theory vol. 104, pp. 102127, 2020.

[9] B.K. Dewangan, A. Agarwal, M. Venkatadri, and A. Pasricha, "Autonomic cloud resource management." In 2018 fifth international conference on parallel, distributed and grid computing (PDGC), IEEE, pp. 138-143, 2018.

[10] I. Odun-Ayo, B. Udemezue, and A. Kilanko, "Cloud service level agreements and resource management." Adv. Sci. Technol. Eng. Syst. vol. 4, no. 2, pp. 228-236, 2019.

[11] S. Mustafa, K. Sattar, J. Shuja, S. Sarwar, T. Maqsood, S.A. Madani, and S. Guizani, "Sla-aware best fit decreasing techniques for workload consolidation in clouds." IEEE Access vol. 7, pp. 135256-135267, 2019.

[12] S.S. Gill, I. Chana, M. Singh, and R. Buyya, "CHOPPER: an intelligent QoS-aware autonomic resource management approach for cloud computing." Cluster Computing vol. 21, no. 2, pp. 1203-1241, 2018.

[13] J.N. Witanto, H. Lim, and M. Atiquzzaman, "Adaptive selection of dynamic VM consolidation algorithm using neural network for cloud resource management." Future generation computer systems vol. 87, pp. 35-42, 2018.

[14] D. Saxena and A.K. Singh, "Workload forecasting and resource management models based on machine learning for cloud computing environments." arXiv preprint arXiv:2106.15112, 2021.

[15] J. Singh and M.S. Goraya, "Multi-objective hybrid optimization based dynamic resource management scheme for cloud computing environments." In 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), IEEE, pp. 386-391, 2019.

[16] M. Ghobaei-Arani, "A workload clustering based resource provisioning mechanism using Biogeography based optimization technique in the cloud based systems." Soft Computing vol. 25, no. 5, pp. 3813-3830, 2021.

[17] S.A. Ali, M. Ansari, and M. Alam, "Resource management techniques for cloud-based IoT environment." In Internet of Things (IoT), Springer, Cham, pp. 63-87, 2020.

[18] S. Goodarzy, M. Nazari, R. Han, E. Keller and E. Rozner, "Resource management in cloud computing using machine learning: A survey." In 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, pp. 811-816, 2020.

[19] D.P. Sharma, B.K. Singh, A.T. Gure, and T. Choudhury, "Emerging Paradigms and Practices in Cloud Resource Management." In Autonomic Computing in Cloud Resource Management in Industry 4.0, Springer, Cham, pp. 17-39, 2021.

[20] M.R. Raza, and A. Varol, "QoS parameters for viable SLA in cloud." In 2020 8th International Symposium on Digital Forensics and Security (ISDFS), IEEE, pp. 1-5, 2020.

[21] M. Daraghmeh, S.B. Melhem, A. Agarwal, N. Goel, and M. Zaman, "Linear and logistic regression based monitoring for resource management in cloud networks." In 2018 IEEE 6th international conference on future internet of things and cloud (FiCloud), IEEE, pp. 259-266, 2018.

[22] S.R. Swain, A.K. Singh, and C.N. Lee, "Efficient Resource Management in Cloud Environment." arXiv preprint arXiv:2207.12085, 2022.

[23] M.H. Khalil, M. Azab, A. Elsayed, W. Sheta, M. Gabr, and A.S. Elmaghraby, "Auto resource management to enhance reliability and energy consumption in heterogeneous cloud computing." International Journal of Computer Networks & Communications vol. 12, no. 2, 2020.

[24] R. Yadav, W. Zhang, O. Kaiwartya, P.R. Singh, I.A. Elgendy, and Y-C. Tian, "Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing." IEEE Access vol. 6, pp. 55923-55936, 2018.

[25] R. Mandal, M.K. Mondal, S. Banerjee, and U. Biswas, "An approach toward design and development of an energy-aware VM selection policy with improved SLA violation in the domain of green cloud computing." The Journal of Supercomputing vol. 76, no. 9, pp. 7374-7393, 2020.

[26] T. Khan, W. Tian, G. Zhou, S. Ilager, M. Gong, and R. Buyya, "Machine learning (ML)–Centric resource management in cloud computing: A review and future directions." Journal of Network and Computer Applications pp. 103405, 2022.

[27] F. Zaker, M. Litoiu and M. Shtern, "Formally Verified Scalable Look Ahead Planning for Cloud Resource Management." ACM Transactions on Autonomous and Adaptive Systems (TAAS) 2022.

[28] M.A.N. Saif, S.K. Niranjan, and H.D.E. Al-Ariki, "Efficient autonomic and elastic resource management techniques in cloud environment: taxonomy and analysis." Wireless Networks vol. 27, no. 4, pp. 2829-2866, 2021.

[29] M.O. Agbaje, O.B. Ohwo, T.G. Ayanwola, and O. Olufunmilola, "A Survey of Game-Theoretic Approach for Resource Management in Cloud Computing." Journal of Computer Networks and Communications vol. 2022, 2022.

[30] S. Mustafa, K. Bilal, S.U.R. Malik, and S.A. Madani, "SLA-aware energy efficient resource management for cloud environments." IEEE Access vol. 6, pp. 15004-15020, 2018.

[31] R. Rajavel and M. Thangarathanam, "Agent-based automated dynamic SLA negotiation framework in the cloud using the stochastic optimization approach." Applied Soft Computing vol. 101, pp. 107040, 2021.

[32] A. Kannaki, V. Azhagu and J.M. Gnanasekar, "A Novel Multi-Agent Approach to control Service level Agreement Violations in Cloud Computing." Turkish Journal of Computer and Mathematics Education vol. 12, no. 12, pp. 1431-1438, 2021.

[33] A. Heidari, and N.J. Navimipour, "A new SLA-aware method for discovering the cloud services using an improved nature-inspired optimization algorithm." PeerJ Computer Science vol. 7, pp. e539, 2021.

[34] J.K. Samriya, S.C. Patel, M. Khurana, P.K. Tiwari, and O. Cheikhrouhou, "Intelligent SLA-aware VM allocation and energy minimization approach with EPO algorithm for cloud computing environment." Mathematical Problems in Engineering vol. 2021, 2021.

[35] A. Belgacem, S. Mahmoudi, and M. Kihl, "Intelligent multi-agent reinforcement learning model for resources allocation in cloud computing." Journal of King Saud University-Computer and Information Sciences 2022.

[36] S. Son, and S.C. Jun, "Negotiation-based flexible SLA establishment with SLA-driven resource allocation in cloud computing." In 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, IEEE, pp. 168-171, 2013.

[37] L. Wu, S.K. Garg, R. Buyya, C. Chen and S. Versteeg, "Automated SLA negotiation framework for cloud computing." In 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, pp. 235-244, 2013.