A Novel Multi-Agent Approach to control Service level Agreement Violations in Cloud Computing

A.Kannaki @ Vasantha Azhagu[1], Dr. J.M. Gnanasekar[2]

[1]Ph.D. Scholar, Department of Computer Science and Engineering Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram, Tamil Nadu

[2]Professor, Dept. of Computer Science and Engineering, Sri Venkateswara College of Engineering Pennalur, Sriperumbudur.

[1]kannakianbu@ymail.com,[2] jmg_sekar@yahoo.com

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

Abstract: Cloud computing becomes popular platform since it offers all key resources from infrastructure to software as on-demand service to the customers through internet. Moreover, its virtualized physical resources are dynamically scalable to meet any surge in the workload. Cloud Service Providers (CSPs) give assurance to quality of their services based on their customer requirements in the form a contract called Service Level Agreement (SLA). However, SLA violations affect the business operations of both CSP and cloud customers as CSP providers must compensate the loss of their customer. This paper presents new agent based architecture which minimizes the SLA violation automatically and hereby improves the trust of all stakeholders**.** It uses multiple many agents like coordinating agent, negotiation agent and arbitrator agent to control the SLA violations. The performance of the proposed work is assessed using CloudSim and Java Agent DEvelopment Framework (JADE). The experimental results reveal that the proposed multi agent based approach performs better than the existing approaches in terms of profit, penalty, and cost execution time and successful execution rate.

Keyword: Cloud computing; Service Level Agreement; Multi Agent; SLA Violations

1. Introduction

Cloud computing become the de-facto platform to deploy resource intensive applications due its unique features like high-performance, scalability and automatic resource provisioning. It inherits the key aspects of many technologies including Virtualization, Service Oriented Architecture (SOA) and utility computing[14]. Cloud computing creates a shared pool of IT resources to support varying workloads of all kinds of applications. It provides virtualized IT resources as services, namely: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) to end users. Many companies including Google, Amazon and Microsoft have used cloud services and provide cloud services to their customers through their data centers. Perhaps, cloud computing has been widely accepted due to its flexibility and pay as you go model.

Cloud Service Providers (CSPs) assure to provide services at a predefined standard, called Service Level Agreement SLA) for their customers. Basically, SLA is a legal agreement signed between the service provider and the consumer which covers the level of expectation between the parties. SLA covers various aspects of services including Quality of Service (QoS) parameters, service deliverability, and penalties for SLA violations. CSPs use a Cloud Management Systems to ensure the quality of services as per the SLA and retain the trust of cloud customers. . Since SLA violations severely affect the reliability and trustworthiness, CSPs take utmost care to optimize the number of SLA violations through numerous methods.

SLA is a formal agreement which defines the quality of all Non-Functional Requirements (NFRs) associated with a service. SLA specifies the various Service Level Objectives(SLOs) and the Key Performance Indicators(KPIs) to verify whether the Quality of Service(QoS) has met the customer expectations or not. Moreover, SLA includes the details of penalties for the service providers in case of SLA violations. Since cloud services are offered to large number of customers, it is difficult to assurance to meet the QoS in all circumstances and detect SLA violations. Hence, it is necessary to automate both controlling and monitoring the quality of services within the negotiated terms.

Many techniques have been proposed to automate for resource managemtn through SLA negotiation and enforcement. These methods usually allocate cloud resources to users by combining virtualization and market based resource allocation policies [STAR]. Many efforts have been taken to automate the SLA aware resource allocation in the recent years. In the literature, few methods focus on framing SLA through negotiation while others are used to automate the enforcement of SLA. However, very few techniques have attempted to support the self-management of cloud services according to the dynamic QoS requirements of cloud users using multi-agents. Moreover, there is a need for new techniques to optimize the SLA violations through reaching a feasible SLA through negotiation before the service deployment on the cloud. Nevertheless, the SLA should be flexible to meet the dynamic requirements of resources during the execution of applications with respect to the workload.

This paper proposes a multi-agent based framework which can be used to both monitor and enforce the pre-defined SLA. The proposed method will optimize the SLA violations by defining the Service Level Specification (SLS) with the involvement of various agents. It uses a negotiation agent which interacts with multiple providers to find the ideal service provider for delivering service without human involvement. It also employs different agents to monitoring the implementation of SLA. It significantly reduces the penalty cost by minimizing the number of SLA violations. The performance of the proposed framework is measured using CloudSim and JADE. The experimental results reveal that it reduces the SLA violations significantly when compared to the existing frameworks.

The rest of the paper is organized as follows. Section 2 provides the outline of existing framework to minimize the SLA violations. Section 3 describes the proposed framework using multiple agents. Section 4 provides the simulation results and Section 5 concludes the paper.

2. Related Work

In cloud computing, many problems are resolved using multi agent base systems. This section presents the outline of some multi agent based approaches to minimize the SLA violations. Yao et al have proposed a multi-agent based system to establishing SLA through the negotiation between the autonomous agents. They have modeled each customer and provider using a set of agents. They have also proposed a negotiation protocol which is used to define the end to end QoS requirements automatically within their framework. However, their framework does not focus on monitoring the implementation of SLA and SLA violations. Sam et al have proposed an multi-agent approach the establishment of dynamic SLA through negotiation among the cloud stakeholders such as customers, brokers and providers. Unlike the negotiation protocol in [4], their negotiation protocol uses various strategies to support concurrent negotiation between the customer and multiple service providers using coordination algorithms. Since their complex cloud negotiation mechanism multiple cloud service providers, it is more useful in the interrelated cloud resource markets.

Gutierrez et al have proposed dynamic agent-based service composition algorithm to optimize the SLA violations[7]. They have tested the performance of their proposed multi agent approach using the colored and object Petri nets. Their approach is capable to handle concurrent and parallel service composition and offer heterogeneous cloud services. Núñez et al have proposed a multi-agent based framework called MASCloud which aims to find the cost-effective configuration to run the customer applications[6]. They have achieved it using mng-agents and sim-agents. In their framework, sim-agents are used to simulate different cloud models offered by providers whereas mng-agents are used to manage and deploy the application in the most appropriate cloud model.

Saurabh Deochake and Debajyoti Mukhopadhyay have proposed an agent based belief–desire–intention (BDI) model to negotiate QoS terms automatically[8]. The key aspect of this approach is that it uses dynamic deadline algorithms to generate offers and counter offers of different cloud service providers as in the cloud market place. Moreover, it computes and assigns the Behavior Norm Score(BNS) and Reputation Index(RI) of each agent to indicate their trustworthy. Maarouf et al have used multi-agent systems which use the Model Driven Engineering(MDE) to describe the SLS (Service Level Specification) in SLA[9]. They then used various agents to monitor the enforcement of SLA automatically. However, they did not use agents to create the SLA to meet the dynamic QoS requirements of customers.

Son and Jun proposed an SLA negotiation mechanism which uses an interactive negotiation mechanism to establish more flexible SLA between customer and provider[10]. Their mechanism could negotiate on multiple issues about time slot, cost and SLA violations. They have proved that their approach improve the profit of providers and decreases the SLA violations. Wu et al proposed negotiation framework which automatically defines the SLA between SaaS customer and multiple providers[11]. They also introduced bilateral bargaining and counter offers in the framework to improve customer satisfaction. This facilitates on-stop-shop for the customers to avail a service at desired efficiency.

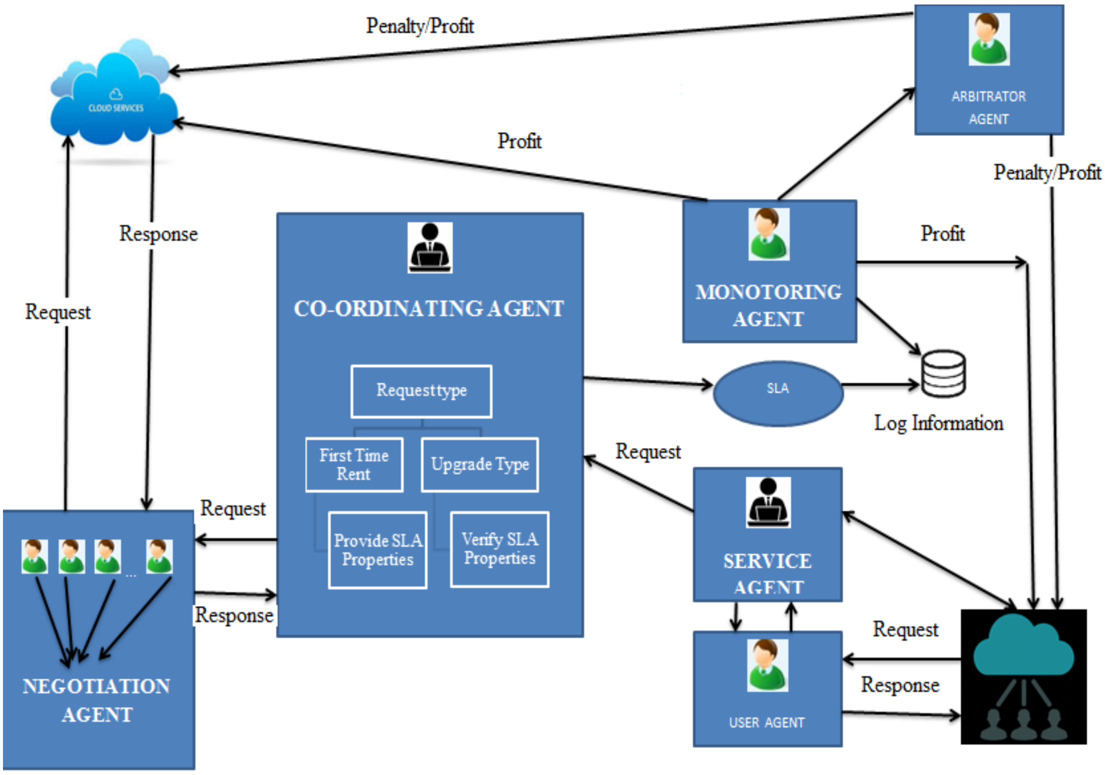
Damian et al have proposed a model named SLA aware Service (SLAaaS) which is orthogonal to other cloud model such as SaaS, PaaS and IaaS[12]. They have developed a domain specific language to describe QoS based SLA of a cloud service using a Cloud Service Level Agreement (CSLA) language unlike the earlier techniques.

They have applied their method on different applications such as e-commerce applications and proved that it will improve key QoS metrics such as performance, cost and dependability.

Sukhpal Singh et al have proposed a agent based method called SLA-aware autonomic Technique for Allocation of Resources (STAR) to manage cloud resources automatically with the indent to reduce the number of SLA violations. This method also optimizes other key QoS parameters execution time, latency, reliability and cost to improve the cloud customers satisfaction. They have tested the performance of the proposed method on the real cloud environment and proved that their method has significantly reduces the SLA violation rate as compared to the Proactive Resource Allocation (PRA) technique and Partial Utility-driven Resource Scheduling (PURS) technique.

3. Architecture of the Proposed Framework

The proposed framework uses multiple agents for establishing, enforcing and monitoring the SLA between the consumer and Cloud Service Provider. The Figure.1 shows the architecture of the proposed framework which establishes the SLA through negotiation without human intervention and imposes penalties in case of SLA violations. The framework uses six intelligent agents, namely: These User Agents (UAs), System Agents(SAs), Negotiation Agent (NA), Coordinating Agent(CA), Monitoring Agent (MA) and Arbitrator Agent(AA). The roles and responsibilities of each agent are as follows.



User Agents: The cloud service consumer requests the cloud server for a service through the User Agent. The UA is responsible to complete the registration of new users. It collects all details about the service from the consumer and forwards the service details to the system agent. Since each consumer is mapped to a single user agent, the proposed framework does not create multiple user agents for additional services by the same user.

System Agents: Upon receiving the request from the user agent, the system agent represents the request in technical terms such as mentioning the QoS factors like product type, Account Type, Contract Length, number of accounts, Response time and solution time. It then verifies the type of service and sends the message to either negotiation or coordinating agent.

Negotiation Agent: The Negotiation process is initiated between the service provider and user agent by considering the various technical factors such as service nature, service reliability, response time and resolution, monitoring, service reporting and Liabilities. The negotiation agent communicates with different service providers. Service Providers submit the bids based on their resource availability, resource capability, business objectives and market circumstances. Unlike the existing negotiation frameworks, the proposed system publishes the offer to all service providers[15-19]. Hence, there is a possibility to get a more feasible SLA with less cost is ensured. It then identifies and selects the best provider whose attribute suits the best will be identified and selected. The selected service along with the service provider is now directed to the coordinating agent.

Coordinating Agent: Once the coordinating agent receives the message from the negotiation agent. it checks whether the message is the request to access the service for the first time or it is related to service upgradation. After the type analysis, SLA is formally established between the service provider and service customer and the same is sent to both the parties. It is also preserved for enforcement. The SLA is then forwarded to the monitoring agent.

Monitoring Agent: After the establishment of SLA between parties, the monitoring agent continuously monitor for SLA violations. Once it finds a violation, it sends the indication to the Arbitrator agent for imposing penalties. If no violation is detected on successful completion of task, the agent sends the message for profit recommendation to the concerned service provider.

Arbitrator Agent: When the Arbitrator Agent (AA) receives the message from the monitoring agent, it analysis the type of violation and its causes. It then enforces penalties based on the SLA on service provider or customer.

4. Experimental Results

The performance of the proposed method is evaluated using the cloud environment developed using CloudSim and Java Agent Development Environment (JADE)[20-22]. As JADE follows Foundation for Intelligent Physical Agents (FIPA), it is widely used to create software agents. The entire cloud framework components are implemented as agents and messages as defined in FIPA are exchanged between the agents for collaboration. Apart from the creation of User Agents (UAs), System Agents(SAs), Negotiation Agent (NA), Coordinating Agent(CA), Monitoring Agent (MA) and Arbitrator Agent(AA), cloud broker, cloud registry, resource manager , physical machine manager and virtual machine manager are also created. We have tested the framework by varying the execution time, workload and number of resources. The simulation assumes that there are four service providers in the market. In our simulation, Virtual Machine requests are sent to the service broker. The resultant values are an average of values after running each experiment for 20 times. The proposed framework is compared with PRA[10] and PURS [11] that are used to manage clouds. The performance of the proposed MAS based, resource management method is compared with Proactive Resource Allocation (PRA) and Partial Utility driven Resource scheduling techniques. All these techniques are used to reduce the SLA Violation rate in the cloud. The SLA Violation Rate(SVR) is defined as the product of service failure rate and the weight of the SLA as in the equation 1.

SLA Violation Rate = Failure Rate(FR) X Weight --- (1)

where, Failure Rate(FR) is calculated as the ratio of failure of the workload to the total number of workloads as given in equation 2.

Failure Rate (FR) = Failure Rate of the Workload --- (2)
Total Workload

The SVR is measured by varying the number of workloads and number of resources of the cloud. The SVR is also computed for different execution time of various workloads in the cloud. The execution time of a workload is calculated as follows

Execution Time$_i$ = Work Completion Time$_i$- Work Submission Time$_i$ --- (3)
Total number of Workload

The energy consumption of the all the frameworks are also computing using the following equation 4 as below:

Energy Consumption = (k*max)+(1-k)*max*$E_{vm\ I}$ --- (4)

Where k is set to 0.5 and is the energy consumption of virtual machine to run the application. The total energy consumption is the sum of energy consumption of the entire workload. All the experiments are done with workload name, CPU, RAM, HDD and Bandwidth requirements and deadlines. The SLA violation rate of the proposed multi-agent method is compared with the existing frameworks. As shown in the figure.2, the SVR of the proposed method is significantly reduced as compared to the PRA and PURS when the execution time of workloads is increased. Both the proposed and PURS method exhibit the same behavior when the execution time is less than 50. However, the proposed method performs much better when the execution time increases.

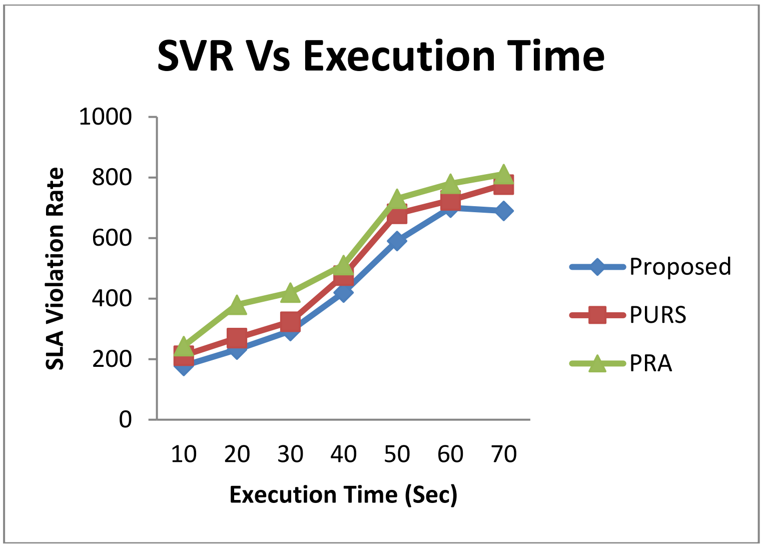


Figure 2. SVR Vs Execution Time

The total available resources in the cloud are varied to find out its impact on the SLA violation rate. As compared to the PRA, PURS and the proposed method has significantly minimized the number of SLA violations when the number of resources is increased. It is also evident that the SLA violation rate is very low unlike the PURS technique.

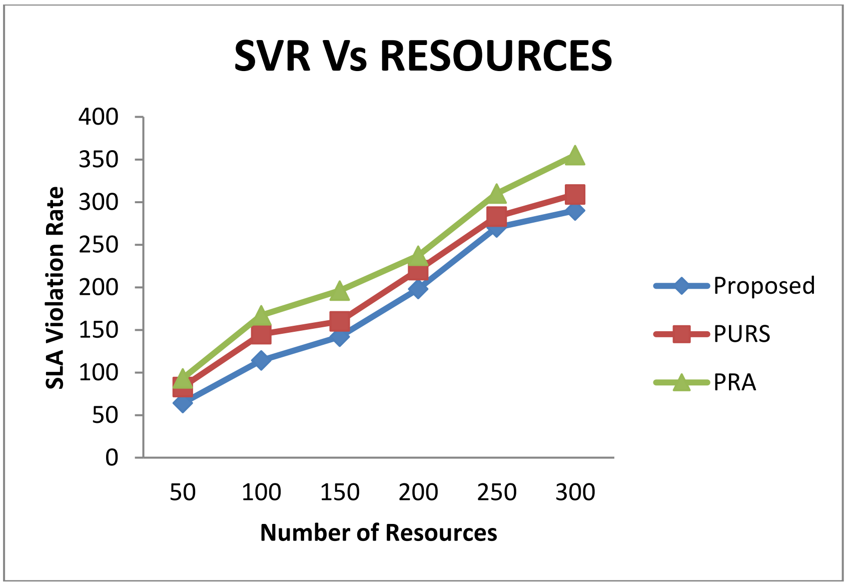


Figure 3. SVR Vs Resources

The SVR of all three methods have been assessed by varying the workload in the cloud. The empirical results reveal that the proposed method outperforms the existing frameworks when the number of workload is increased. The experiments also show that PRA has high SLR compared to the other two frameworks.

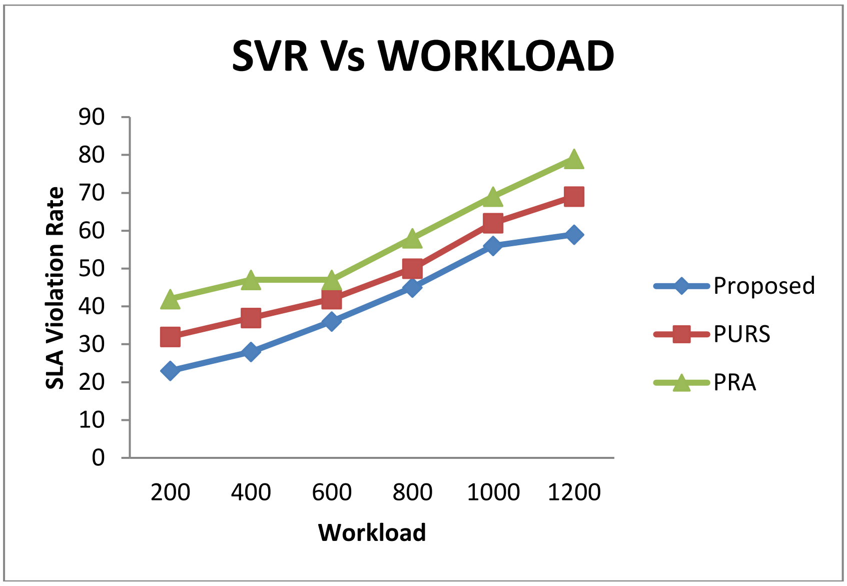


Figure 4. SVR Vs Workload

The energy consumption of the frameworks is measured using the simulator. As shown in the figure.5, the overall energy consumption is increased in line with the workload irrespective of the framework. However, the proposed method intelligently utilizes the cloud resources to reduce the energy consumption as compared to PRA and PURS.

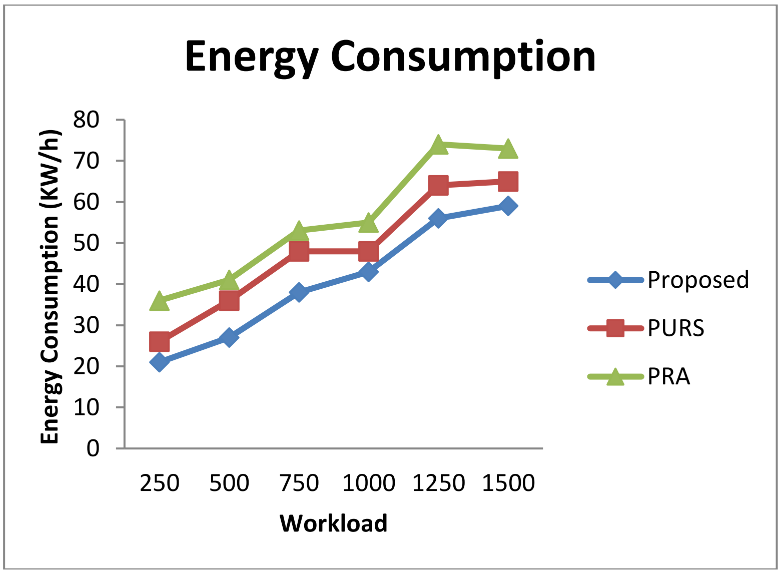


Figure 5. Energy Consumption

5. Conclusion

Since the success of adopting cloud computing lies on the enforcement of SLA, it is important to establish the flexible and feasible SLA between the cloud stakeholders through negotiation. This paper proposes the unified framework which is used to establish, monitor and manage the SLA using multi-agent based system. The proposed system automates the implementation of SLA and optimizes the SLA violations using autonomous agents unlike the existing frameworks. The key aspect of the proposed method is that it can be used to provide SLA as a Service with other cloud services such as IaaS, PaaS and SaaS over federated cloud. The performance of the proposed system is assessed through simulation. The results show that it significantly reduces the number of SLA violations when compared to the existing methods. In the future, the impact of SLA violations on QoS parameters will be analyzed.

References

1. D. Talia, “Clouds meet agents: Toward intelligent cloud services,” IEEE Internet Computing, vol. 16, no. 2, pp. 78–81, 2012.
2. M. Schumacher, “Multi-agent systems,” Objective Coordination in Multi-Agent System Engineering: Design and Implementation, pp. 9– 32, 2001.
3. J. Yan, R. Kowalczyk, J. Lin, M. B. Chhetri, S. K. Goh, and J. Zhang, “Autonomous service level agreement negotiation for service composition provision,” Future Generation Computer Systems, vol. 23, no. 6, pp. 748–759, 2007
4. J. Yan, R. Kowalczyk, J. Lin, M. Chhetri, S. Goh, and J. Zhang, “An agent negotiation approach for establishment of service level agreement,” in Computer Supported Cooperative Work in Design III, ser. Lecture Notes in Computer Science, W. Shen, J. Luo, Z. Lin, J.-P. Barths, and Q. Hao, Eds. Springer Berlin Heidelberg, 2007, vol. 4402, pp. 459–468. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72863-4 47
5. K. M. Sim, “Towards complex negotiation for cloud economy,” in Advances in Grid and Pervasive Computing. Springer, 2010, pp. 395–406.
6. A. Nunez, C. Andr´es, and M. G. Merayo, “Mascloud: a framework based on multi-agent systems for optimizing cost in cloud computing,” in Computational Collective Intelligence. Technologies and Applications. Springer, 2012, pp. 436–445.
7. J. O. Gutierrez-Garcia and K.-M. Sim, “Agent-based service composition in cloud computing,” in Grid and Distributed Computing, Control and Automation. Springer, 2010, pp. 1–10.
8. Saurabh Deochake and Debajyoti Mukhopadhyay,” An Agent-Based Cloud Service Negotiation in Hybrid Cloud Computing”, DOI: 10.1007/978-981-15-8289-9_55
9. A. Maarouf, M. E. Hamlaoui, A. Marzouk and A. Haqiq, "Combining multi-agent systems and MDE approach for monitoring SLA violations in the Cloud Computing," 2015 International Conference on Cloud Technologies and Applications (CloudTech), Marrakech, Morocco, 2015, pp. 1-6, doi: 10.1109/CloudTech.2015.7336975.
10. S. Son, and S. C. Jun, “Negotiation-based flexible SLA establishment with SLA-driven resource allocation in cloud computing,” in Proc. 13th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput., 2013, pp. 168–171.
11. L. Wu, S. K. Garg, R. Buyya, C. Chen, and S. Versteeg, “Automated SLA negotiation framework for cloud computing." In Proc. IEEE/ACM 13th Int. Symp. Cluster Cloud Grid Comput., 2013, pp. 235–244.
12. Serrano, Damián & Bouchenak, Sara & Kouki, Yousri & Alvares, Frederico & Ledoux, Thomas & Lejeune, Jonathan & Sopena, Julien & Arantes, Luciana & Sens, Pierre. (2015). SLA guarantees for cloud services. Future Generation Computer Systems. 54. 10.1016/j.future.2015.03.018.
13. S. Singh, I. Chana and R. Buyya, "STAR: SLA-aware Autonomic Management of Cloud Resources," in IEEE Transactions on Cloud Computing, vol. 8, no. 4, pp. 1040-1053, 1 Oct.-Dec. 2020, doi: 10.1109/TCC.2017.2648788.
14. R. Buyya, C. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: vsion, hype, and reality for delivering computing as the 5th utility. Future Gener. Comput. Syst. 25, 599–616 (2009)
15. Dewangan, B.K., Agarwal, A., Venkatadri, M., Pasricha, A. (2019). Design of self-management aware autonomic resource scheduling scheme in cloud. International Journal of Computer Information Systems and Industrial Management Applications, 11: 170-177.
16. V. Stantchev, C. Schröpfer, Negotiating and enforcing QoS and SLAs in grid and cloud computing, in Advances in Grid and Pervasive Computing, pp. 25–35 (2009)
17. S. Venticinque, R. Aversa, B. Di Martino, M. Rak, D. Petcu, A cloud agency for SLA negotiation and management, Euro-Par 2010 Parallel Processing Workshops., pp. 587–594 (2011)
18. B. Shojaiemehr, A. Rahmani, N. Qader, A three-phase process for SLA negotiation of composite cloud services. Comput. Stan. Interfaces 64, 85–95 (2019)
19. J. Sim~ao and L. Veiga, “Partial utility-driven scheduling for flexible SLA and pricing arbitration in clouds,” IEEE Trans. Cloud Comput., vol. 4, no. 4, pp. 467–480, Oct.-Dec. 1, 2016.
20. K. M. Sim, “Agent-based cloud computing,” Services Computing, IEEE Transactions on, vol. 5, no. 4, pp. 564–577, 2012
21. M. Schumacher, “Multi-agent systems,” Objective Coordination in Multi-Agent System Engineering: Design and Implementation, pp. 9– 32, 2001.
22. J. Yan, R. Kowalczyk, J. Lin, M. Chhetri, S. Goh, and J. Zhang, “An agent negotiation approach for establishment of service level agreement,” in Computer Supported Cooperative Work in Design III,

ser. Lecture Notes in Computer Science, W. Shen, J. Luo, Z. Lin, J.-P. Barths, and Q. Hao, Eds. Springer Berlin Heidelberg, 2007, vol. 4402, pp. 459–468.

23. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72863-4 47
24. H. Morshedlou, and M. R. Meybodi, "Decreasing impact of SLA violations: A proactive resource allocation approach for cloud computing environments," IEEE Trans. Cloud Comput., vol. 2, no. 2, pp. 156–167, 2014.