International Journal of Cognitive Computing in Engineering 1 (2020) 27–35

Contents lists available at ScienceDirect

International Journal of Cognitive Computing in Engineering

journal homepage: www.elsevier.com/locate/ijcce

An automated resource management framework for minimizing SLA violations and negotiation in collaborative cloud

K.S. Sendhil Kumar, N. Jaisankar*

School of Computer Science and Engineering, VIT Vellore, Tamilnadu, India

ARTICLE INFO

Keywords:
Collaborative cloud computing
Service Level Agreement (SLA)
SLA violation
SLA negotiation
Hybrid Cuckoo Search

ABSTRACT

Collaborative Cloud computing is an emergent and encouraging computational prototype for sharing on-demand cloud resources in a multi-cloud environment. In a collaborative cloud environment, we make use of unused computational resources within an interconnected network to provide seamless service to the customers. SLA (Service Level Agreement) means a legal agreement signed among the consumer and the cloud service provider to guarantee a better quality of service. Nowadays due to more user request, suppliers of cloud service are not able to deliver guaranteed Quality of Service (QoS) to the customers leading to SLA violation. Existing SLA- based resource allocation models in today's cloud market consider an agreement for each QoS parameters independently and follow fixed pricing models, which results in lacking an end-to-end approach for the client task that requires composing multiple services and inefficient utilization of computing resources, which has direct negative effect on performance and cost. In this paper, we have designed a framework model for the effective dynamic provision of resources considering various categories of SLAs laterally with their penalty. We used Hybrid Cuckoo Search algorithm for scheduling of user requests in collaborative cloud computing environment. The proposed framework services the customer requests from its interconnected cloud servers and has an SLA Negotiation policy to minimize the penalty cost and maximize customer satisfaction by reducing SLA violations.

1. Introduction

Cloud computing turns into one of the most significant platforms for suppliers of cloud service to provide service requests in a virtual manner via the internet to customers (Mell & Grance, 2011). According to Foster et al. (2008) and Wang et al. (2011) cloud computing acts as a backbone for several companies today for the reason of its greater elasticity, user-friendliness, and volume compared to outdated online computation and storage techniques. Cloud customers across the globe interchange their information with a wide variety of computing resources distributed through various service providers that are handling diverse categories of datasets which need to be accessible to various customers with dissimilar access privileges. Hence the current demand for scalable computing capabilities will usually increase among the cloud consumers. As a result single cloud server possibly will unable to discover and associate with a wide variety of capability to the application during execution time. Therefore the scientists are in necessity to construct a virtual atmosphere for communicating various cloud servers (Khan and Zomaya, 2015).

Collaborative cloud offers the base for economical and effective digital transformation. The present most inventive organizations are eager adopters of collaborative cloud. Organizations with substantial amount of work in the hybrid cloud have been capable to implement digital transformation creativities more rapidly and raise their income. Collaborative cloud environment means multiple cloud servers are interconnected together so that resources can be shared between them. If one cloud service is not having sufficient resources then it will pass this information to another cloud server from which we can able to satisfy customer requests. In this way, we can able to provide seamless service to the customer. Multiple cloud environments have many unique features such as resources belonging to various cloud providers are completely distributed, heterogeneous, and totally virtualized (Li et al., 2015).

Collaborative cloud computing has arisen as a promising solution for providing on-demand access to virtualized re- sources, providing platforms, and running applications in a pay-as-you-go manner. According to Hao et al. (2014) collaborative cloud makes use of the unused resources of interconnected clouds to offer the requested services as a result the public gains services from the cloud environment. According to Panda and Jana (2017) more organizations are migrating from traditional in-house infrastructure towards cloud environment for deploying a wide range of business applications. In order to service increasing customer's requests in a better way, cloud service providers are

* Corresponding author at Professor, School of Computer Science and Engineering, VIT Vellore.
E-mail addresses: sendhilkumar.ks@vit.ac.in (K.S.S. Kumar), njaisankar@vit.ac.in (N. Jaisankar).

https://doi.org/10.1016/j.ijcce.2020.09.001
Received 2 February 2020; Received in revised form 21 September 2020; Accepted 21 September 2020
Available online 26 September 2020

migrating to a newly emerging concept of collaborative cloud, where the customers can utilize the resources from multiple cloud service providers.

The main benefit of using the collaborative cloud is customers can able to deploy their application on various cloud service providers. When a cloud service provider is not able to provide the resources demanded by its customer then it will search the availability of resources in its interconnected cloud server provider and utilizes the resources if it is available (de Asís López-Fuentes and García-Rodríguez, 2016). Moving to collaborative cloud computing will offer you lower cost and improved QoS performance. Collaborative cloud offers the following benefits for customers (Hao et al., 2014):

1 Easy way of storing and retrieving of data.
2 Proficient collaboration: permits multiple persons to collaborate on content.
3 Complex collaboration: provides a whole environment which incorporates, storing and retrieving, content management, and office efficiency applications for instance word processing, spreadsheet, presentations, calendaring and workflow.
4 Functional applications: provides practical applications such as project management.
5 Business and social networking: provides the capability for social networking groups to be formed alongside basic storing and recovery of documents

1.1. Service Level Agreements

The key factors for customer satisfaction and generating revenue are Performance, Cost and penalties. These factors have a correlation, which makes the framework more complicated when some key factors are missing. SLA is an preliminary document which talks over the selected parameters as a initial condition for starting any business (Badshah et al., 2020). SLA is considered as an fundamental characteristic for providing scalable resources on-demand in cloud. SLA outlines essential controls for assured Quality of Service (QoS), cost, fault-tolerant capability, security and validity of service. Nowadays, there is a enormous demand for SLA-based scheduling that increases usage of resources on Cloud (Hussain et al., 2019).

According to Suprakash and Balakannan (2019) almost maximum number of service providers are offering the services by means of their Service Level Agreement (SLA) primarily concentrating on to provide best Quality of Service. On the other hand not all the computing capabilities are utilized to maximum extent. The information source of the same from various consumers can be scrutinized thoroughly, categorized and this data can be efficiently utilized to accomplish and renegotiate the services provided. The authors proposes a SLA catalogue based prototype to efficiently use resources of the cloud environment dependent on SLA parameters such as frequency and bandwidth. The SLA catalogue and computing resources are supervised regularly and the resources which are not utilized effectively are transformed to idle state or turned off. This mechanism guarantees optimal resource utilization as well as preserve the quality of service.

According to Zhu et al. (2011), Service Level Agreement (SLA) is a mutual legal agreement signed between the end user and the service provider of cloud which define what kind of service the customer expects from the service provider. Numerous consumers want to execute diverse kinds of applications on the cloud server. So each consumer possibly will have various Quality of Service (QoS) necessities based on their amount of work. This makes provisioning of resources a challenging task. As soon as the service providers are identified, it is essential to discover the different elements of an SLA that will be signed by agreeing metrics. These components are known as terms of service (Aljoumah, 2015). Suppliers of cloud service and the consumer are agreeing upon a certain performance related Quality of Service (QoS) properties, for instance reply time, availability of the resources that must be retained by a cloud provider during execution of the services, which is documented in the form of Service Level Agreements (SLAs) (Zhang et al., 2010). SLAs generally comprise segments to address: services definition, performance measurement, problem management, consumer responsibilities, guarantees, disaster recovery, and agreement termination. SLAs are result based in that their determination is precisely to express what the consumer will obtain (Peng and Gala, 2014). According to Wu and Buyya (2012) the various steps in SLAs are discovering the Service Provider, Define Service Level Agreement, Establish the Agreement, Monitor SLA violation, Terminate SLA, Impose Penalty for SLA Violation.

According to Slimani et al. (2020) one of the significant difficulties for service providers of cloud applications is the way to ensure high accessibility of the provided applications while satisfying users QoS. In order to handle this issues efficiently, reproduction methods are generally applied. According to the used granularity for replication there are two significant ways to attain reproduction: the first approach is through replicating the service and is generally known as Service-oriented Replication (SoR). The second method is replicating the underlying data and is termed as Data-oriented Replication (DoR).

1.2. Role of metrics to support SLA

Effective SLAs are essential to guarantee good business relationship, client fulfillment, and expectation. The measurement used to quantify and accomplish performance promise to SLA guarantees are the core of a fruitful contract and are important for a extended duration success factor. Absence of adequate knowledge and understanding in the usage and computerization of performance metrics raises issues in some companies as they attempt to formulate their SLA techniques and set the measurements required to maintain those procedures (Paschke and Schnappinger-Gerull, 2006).Service level contracts can encompass various service performance metrics with consistent service level objectives. SLAs measure the service provider's performance and quality in a various manners. A few measurements that SLAs may indicate incorporates Availability and uptime - which denotes the fraction of time customers can able to avail the services provided by the service provider, Number of concurrent consumers that can be attended, Explicit standards to compare the real performance from time to time, Application response time, In case of network changes, prior notification has to be given to the user, VM Usage statistics and Help desk response time for different classes of difficulties.

2. Background

In paper Serrano et al. (2016) proposed a novel cloud model named SLA-aware Services (SLAaaS) which integrates various service levels and SLAs into cloud providing better performance and cost reduction on cloud services. This model can be functional in any of the Service models of cloud such as IaaS, PaaS, and SaaS. In this model the authors mainly focused on three works

1 Developing a novel language called CSLA (Cloud-oriented Service Level Agreements) permits describing SLA in any language for the services of cloud.
2 Presented a common control-theoretic approach managing SLA based cloud services.
3 Applying CSLA and control theoretic method provides assured SLA services

Numerous challenges in implementing SLA into cloud environment are 1. How to think about SLA in a general manner for various cloud situations? 2. How to define the SLA contracts among a cloud supplier and a cloud consumer, for instance service levels goals, or fines in case of violations of SLA?, 3. How to offer assurances on cloud Quality of Service?. In this work even though the authors considered cost as a parameter in SLA they have not taken cloud billing model based on resource consumption into account. In our work, we overcome this issue by considering cloud billing model and SLA violations into account.

Existing cloud computing systems needs an improved resource provisioning models to handle rapidly changing enterprise requirements. Traditional models failed to collectively integrate consumer based service management, managing and controlling risk in computation, and management of resources automatically into a market-based resource management system. In this paper Buyya et al. (2011) present their vision, difficulties encountered, and structural components of SLA-based resource management. The suggested design supports a combination of virtualization technologies and various strategies for market-based provisioning of resources to applications.

El-Matary et al. (2019) presents an automatic SLA negotiation structure based on autonomous and user friendly mediators and multi agent systems dependent on the autonomic processing features as appropriate devices for self-identification of vulnerabilities and self-checking for the Cloud processes and amenities. The authors proposed Autonomic Negotiation Layers (ANL) architecture is framed in order to apply the dynamic Cloud Computing process in negotiation of service process and to assemble these activities in a defined negotiation design layer. The suggested automated negotiation framework is dependent on the automatic intellectual agents mainly comprises five agents beside the consumer and the supplier. Later these components are considered as critical modules in generating the SLA agreement.

An SLA-aware management of resources virtually for cloud setups was recommended by (Van et al., 2009) in which the authors developed an automated resource manager for controlling the virtual setup which decouples the allocation of resources from the dynamic scheduling of virtual machines. Despite the fact that the paper satisfies the SLA and cost of operation, it does not deal with concerns linked to SLA penalty. In our proposed work we are concentrating on various SLA negotiation policies and penalty related issues.

Minarolli and Freisleben (2014) presented a distributed Artificial Neural Network centered resource provisioning to streamline the trade-off among the incompatible goals of fulfilling applications QoS necessities and reduced energy costs. Artificial Neural Network (ANN) is used to identify the finest resource provisioning approach to virtual machines that enhance the utility function. The resource manager is in charge of resource allocation process. Every manager builds an ANN based model with only two inputs i.e. CPU and only one VM memory allocation. In order to predict the power consumption of the physical machine in advance, they build an ANN based power model. In the meantime the energy intake of the physical machine relies upon allocation of resources to all Virtual Machines, the amount of inputs to the energy model is twice the number of VMs. These works concentrate on particular applications, not heterogeneous requests from users, which are considered in our proposed work. To offer support for massive number of virtual machines, the authors recommended an SLA-based resource management framework consisting of SLA manager and resource manager responsible for allocating resources and monitoring the QoS services of a single VM.

At present cloud data centers are experiencing the concerns of not efficiently utilizing the resources and acquiring additional charge. They are being utilized to execute various kinds of applications such as a small web based application to a high-performance computing, which have distinct QoS requirements. This creates the issue more problematic because it is hard to compute the volume of the server that has to be assigned to individual VMs. Garg et al. (2014) presented an innovative approach that makes best use of the usage of data center by running various types of user application demands, specifically non-interactive and transactional applications with different SLA requirements. They used an admission control and scheduling mechanism through which maximized resource utilization is accomplished and furthermore assured that the QoS necessities of clients are satisfied as stated in SLAs. Virtual machines are dynamically allocated such that SLA contracted with the client is satisfied without any fine. They tested with dissimilar types of SLAs along with appropriate fines for different user requests for enhanced resource allocation and usage of datacenters. The limitation of this work is this mechanism will be best suitable for single cloud data center while it won't be appropriate for multiple cloud environments. In our proposed work we have introduced a framework model which runs applications in a multi-cloud environment without affecting the request from other applications.

Son et al. (2016) presented an adaptive negotiation approach for establishment of a multi- purpose SLA under fluctuating cloud capabilities. The recommended negotiation system adaptively manages negotiation constraints, which signifies favorites between different SLA disputes under an exchange off, by scrutinizing trends in the workloads. Here the cloud model can move the on-peak load and minimizes violations in SLA with elastic pricing. The author's contribution in this paper includes

1 Designing an adaptive negotiation mechanism for establishing a multi-purpose SLA;
2 Provided a direction for deciding SLA negotiation parameters for fixing a price for cloud services and management of resources
3 Demonstration showing bargaining-based SLA establishment, simplified management of cloud resources, and increased revenue of cloud computing environments.

This paper concentrated mainly on negotiating cloud services in a open cloud marketplace where buyers and suppliers participate in the bargaining process, whereas in our proposed model we have designed an automated SLA Negation framework to take care of this process.

2.1. Open challenges and research gaps

1 Existing algorithms provide a higher throughput and prove to be cost-effective. However, they do not take reliability and availability into account. As a result, it is essential to design an algorithm aimed at enhancing the availability, reliability, and improvement to the QoS in cloud computing environment.
2 Whenever there is an increase in the service request, there is a necessity for an improved resource scheduling algorithm for handling user requests.
3 Existing techniques are best suitable for a single cloud data center while they are not appropriate for multiple cloud environments.
4 An automated penalty cost computation and negotiation mechanism is required, when SLA violation occurs.

3. SLA-based resource management framework

Resource management concerns like SLAs (Service Level Agreements) associated with providing services to millions of users by means of a data center is a difficult task when compared to allocating services for a million consumers to execute on their own individual PCs. There are numerous difficulties involved in SLA based resource scheduling to distinguish and fulfill service requirements based on the preferred utility of consumers. Resource provisioning is the process of mapping of physical resources to various consumers based on their demands. Computing capabilities must be provisioned in a manner so that none of the servers in the cloud is overloaded and entire free computing capabilities in the cloud are utilized efficiently. Resource Provisioning allows optimal allocation of resources among various user requests in a finite amount of time to attain desired quality of service. The resource provisioning decisions must adjust to changes in load as they happen. In order to overcome these restrictions, we propose a innovative dynamic resource management approach that not just make best use of resource utilization by distributing resources amongst various simultaneous applications possessed by different consumers but additionally considers SLAs of dissimilar categories and designed a negotiation model for computing penalty cost if SLA violation occurs.

Fig. 1 shows the proposed SLA based resource management framework. Initially, to utilize the services offered by the service provider

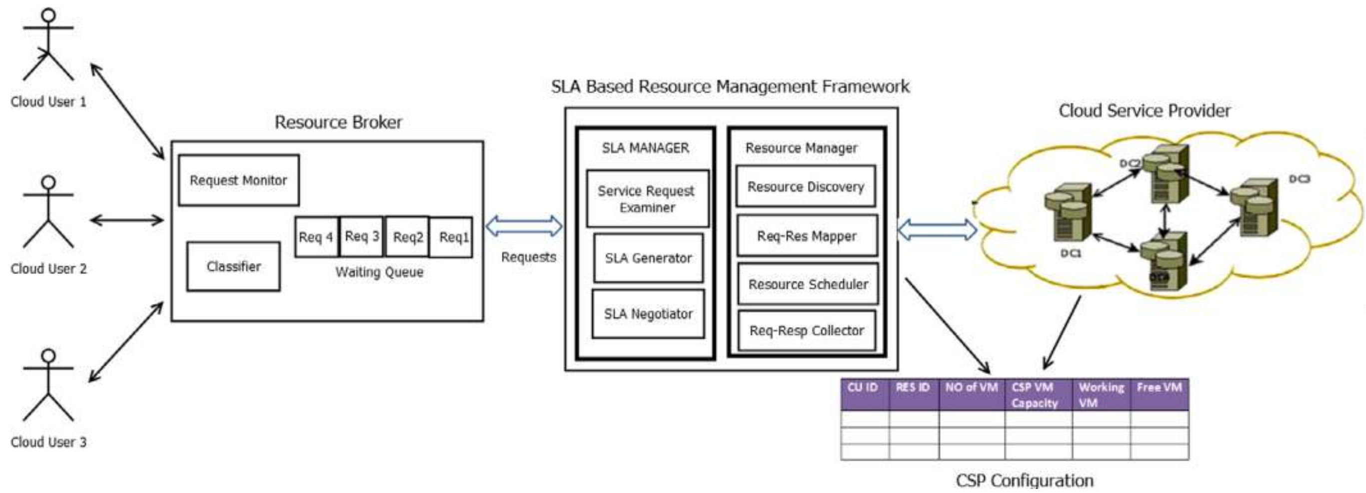

Fig. 1. SLA-based resource management framework.

the cloud user has to register with the resource broker using the web portal by filling up the forms during the registration process. Users profile information such as username, password, Name of the organization, Size of the organization, Type of organization, credit card account, machine IP address, etc. are collected. Once the registration process is completed the user can submit their request to the resource broker. User requests consist of details such as Configuration of VMs, No of VMs required, Type of Operating systems, necessary software required, No of days needed etc. This user request information is passed to the service broker. Upon receiving the customer request the broker will calculate the availability of resources, expected processing time and cost to offer the services. This information is then passed to the service provider. If the service provider cannot fulfill the customer request the broker will search for some other service provider satisfying the user request. Once the service request is satisfied an SLA defining various terms and conditions to be satisfied is established and resource provisioning takes place among the consumer and cloud service provider. Advantages of this framework includes Enhanced customer satisfaction, Improved Quality of Service and maintaining a good relationship between customer and service provider. Our SLA Model uses Hybrid Cuckoo Search scheduling policy for forecasting the SLA penalty charges for individual client requests under various scheduling of resources.

3.1. SLA manager

The responsibility of the SLA manager is to make sure that the entire Service Level Agreements (SLAs) for various Client service requests are delivered according to specifications. SLA manager is responsible for keeping track of SLAs fulfillment among the consumer and service provider. It is the responsibility of the SLA manager to serve as a point of contact between the client and the cloud service provider; manage any escalation process; make any appropriate changes and delivery; review and keep track of services continuously; facilitate and contribute solutions for solving any disputes and evaluate and submit report on the relationship between the customer and service provider on regular basis.

Crucial duties of an SLA manager incorporates Establishment, Overseeing and regulating administration preeminent techniques, Monitoring the performance of SLA for consistence with client needs, smooth communication between cloud service provider and consumers to describe and retain the SLAs necessary for each service and Finally, the SLA manager needs to persistently review the established services and track any occurrences that possibly will delay the smooth delivery of services to the customers. SLA manager consists of components such as

1 Service request examiner
2 SLA generator
3 SLA negotiator

3.1.1. Service request examiner

Resource Broker on behalf of users submits the user's requests using web applications from any part of the world. As soon as a service demand is received, first the Service Request Examiner (SRE) authenticates and analyze the user's QoS requests before taking the decision whether to agree or dismiss the request. SRE guarantees that none of the resources are overloaded whereby several service requirements cannot be satisfied effectively because of restricted accessibility of resources.

3.1.2. SLA generator

SLA Generator formalizes an SLA which gives assurance to the customers' human rights and relate their terms through composing the SLOs to connect the space among the QoS hype and SLA reality. Here an agreement is signed tween customer and cloud service provider and includes service level objectives according to the QoS specified by the customer and cloud service provider. SLA generator defines the Service parameters that are required by the customers. It will clearly mention the Service Level Objectives (SLO) according to the QoS parameters. Roles and responsibilities of the business party's involved including customer and the service provider are clearly defined and the essential tasks to guarantee that the demanded level of services is delivered is taken care. SLA Generator will define the deliverable and non-deliverable services that are covered in the SLA, Penalty and Negotiation policies and SLA validity period.

3.1.3. SLA negotiator

As the Cloud customers do not know the procedure of validating the SLA, it is essential to have a reliable third party who can take care of SLA authentications and guarantee. Several consumers are yet doubtful about Cloud suppliers' QoS guarantees as a result of the space among these assurances and the SLAs offered by these providers. The role of SLA Negotiator is to reduce the gap and provide a confidence among the cloud service supplier and the consumer. SLA Negotiator will applaud consumers to succeed the essential service proficiently after making the best possible decisions for negotiation with multiple service providers. If any SLA violation happens during service provisioning penalty has to be enforced for SLA violation.

3.2. SLA violation and negotiation

In cloud computing environment many factors such as environmental changes, S/W failures, Network performance, Bandwidth etc. may influence the system behavior. SLA violation is a most important characteristic of cloud computing and the violation leads to the decline of consumer fulfillment level and additionally disturbs the cloud supplier leading to punishment or fine. SLA violation can happen under various circumstances such as Performance is provided at the lower level than the agreed, Services provided at the appropriate level but with more delay, Variation in VM resource usage and Services not at all provided.

Negotiating Service Level Agreements (SLAs) plays a vital part in the utilization of services. During negotiation process both the parties, consumers and the providers of the service are united through a collaborative mechanism identified by the process of bargaining. This negotiation process decides the scope of movement of data which in turn impacts convergence upon settlement. A negotiation mechanism decides the cardinality of both the parties taken part in the negotiation process, their responsibilities, the visibility of the bargains traded, administration of a particular session, boundaries for negotiation process, etc. SLA negotiation is an essential system to ensure the performance of cloud service and improve the confidence among cloud service consumers and service providers of cloud. An SLA negotiation among Cloud parties provide assistance in outlining the Quality of Service (QoS) requests of critical service-based processes. Total VM Cost Computation is computed as follows

$$VM\ Cost = CSP_{instcost}\ X\ (ReqRAMSize/0.5) \tag{1}$$

$$Total\ RAM\ Cost = No.\ of\ days\ X\ RAM\ Cost \tag{2}$$

$$Total\ VM\ cost = Total\ RAM\ Cost\ X\ No.\ of\ VM \tag{3}$$

Where *CSPinstcost* is cloud service provider's RAM initial cost. ReqRAMSize is cloud user requested RAM Size. *TotalRAMCost* is Single Virtual Machine Cost. No. of VM is a number of virtual machines needed by the cloud users. *TotalVMCost* is Total Virtual Machine Cost for a cloud user.

If SLA violation occurs penalty has to be enforced for SLA violation. In proposed work two kinds of negotiation policy are adapted. The first method is Refunding the amount with penalty cost being negotiated and the second method is Extension of service provided to the customer. In first method, if the customers do not want to continue the services offered by the service provider, customer can request for refund of amount along with penalty cost. Penalty cost is computed as follows:

$$SVMC = TC/No\ of\ VM \tag{4}$$

Where *SVMC* is Single Virtual Machine Cost. *TC* is Total Cost of VM purchased by a cloud user. No of VM is a number of virtual machines purchased.

$$MPC = ((SVMC\ X\ SLA)/100) \tag{5}$$

Where *MPC* is Minimum Penalty Cost. The value of SLA will vary depending on the service provider.

$$PC = ((MPC\ +\ SVMC)\ X\ RVM) \tag{6}$$

Where PC is Penalty Cost. RVM is Reduced Virtual Machine (i.e. Number of VMs cancelled by the cloud service provider). In second method customer need the service from the same service provider. For SLA violation the service will extend the service as a penalty for additional days apart from actually agreed one as follows:

$$No.\ of\ days\ Extended = Penalty\ cost/VM\ Cost$$

3.3. Resource manager

Resource manager provides a new way to deploy, manage and monitor the customer requests and offers a set of resources and operations for working with the resources that are deployed. Resource Manager is responsible for allocation of resource and consists of several components like Resource Discovery, Req-Resource Mapper, Resource Scheduler and Request Response Collector. Resource Discovery is in charge for identifying cloud service provider which can fulfill the requests of the consumer. Req-Resource Mapper looks the CSP configuration table and checks whether the resources requested by the user are available or not. If it is available it maps the user requirement with the available resources in cloud service provider and allocates it to the requesting user. Resource Scheduler uses a Hybrid Cuckoo Search algorithm to identify which cloud service provider can provide efficient service based on the requirements of the user. It helps to allocate appropriate resources to a right application at right time. So that the applications can use resources effectively. Request Response Collector will collect the responses from the cloud service provider and offer the service to the consumers.

3.4. Hybrid Cuckoo Search scheduling policy

Algorithm 1 shows the Pseudo code of Hybrid Cuckoo Search algorithm. The algorithm begins with the initial values of VM Req of size n, probability pa c [0, 1], the maximum number of iterations Maxitr and the initial iteration counter *t*. Based on the client request the request for virtual machines is grouped from the database. The initial VM Req n is orderly fetch from database and each solution x_i in the VM Req is calculated by computing its fitness function. Employed Bees (BE) compute cloud user request time, Onlooker Bees (BO) compute cloud service provider response time. This fitness value is given as an input to Scout Bees (BS) which finds the best fitness value. Once the fitness value is computed the ordered information is passed to cuckoo search algorithm. Cuckoo search algorithm will repeat the following process until it finds the best service provider for the client request.

Algorithm 1**:** Pseudo code of Hybrid Cuckoo Search algorithm

Fig. 2 demonstrates the Pseudo code for the proposed Hybrid Cuckoo Search algorithm.

Step 1 A new solution is orderly fetched from the database.

Step 2 If its objective function value is improved than the objective function value of the chosen order solution, then the novel solution is interchanged with an earlier chosen solution.

Step 3 A fraction $(1 - p_a)$ of the solutions is orderly chosen, abandoned and interchanged by novel solutions produced via local VM Req Database as follows.

$$X_i^{t+1} = X_i^t + \gamma\left(X_j^t - X_k^t\right) \tag{7}$$

Where $X_j^t$and X_k^t are two dissimilar solutions chosen randomly and γ is a random number. X_i^{t+1} is the

Best Solution of VM Req. *i,j,k* are the index values of iteration.

Step 4 The obtained solutions are rated according to their objective values, then the better solution is allocated and the iteration counter increments.

4. Experimental results and discussions

We can implement the Proposed SLA framework in real cloud environments also. But Nowadays more number of companies are migrating their business and operations to cloud environment and innovative and numerous sophisticated algorithms are being developed. Creating an environmental setup and executing these experimentations in actual cloud infrastructure are costly and can prove to be catastrophic for the entire project as the flaws in algorithms or structures are found after the deployment. However, modeling and simulation tools are appropriate choices that regularly provide better alternative option for imitating cloud computing infrastructures. Moreover, simulating a cloud environment can reduce the cost, time and effort required for setting the real time cloud scenario.

Existing grid-based simulation tools were not able to isolate the cloud services properly. To avoid these various cloud simulation tools are used by the researchers. In our work we are using Cloudsim simulator to model and simulate the SLA model and analyze the performance of the application service. Cloudsim is an extensible simulator that affords basic classes for defining users, computational resources, virtual

```
1. Fetch the total no. of hosts (Data Center) as Tn

2. Get the Initial position of the Bee as Pi

3. While Tn ≠ NULL

      Cluster the VM Req based on client request.

4. For each Clustered Host (Hc) Generate
            4.1 BE – Compute cloud user request time (VMReq_Time)
            4.2 BO- Compute Cloud Service Provider response time (VMRes_Time)
            4.3 BS- Compute the fitness (VMReq_Time,VMRes_Time)

5. BS Find Best Fitness value (Fi) and pass the ordered information to cuckoo.

6. While (T < Maximum Generation)
            6.1 A new solution (Fj) is orderly fetched.
            6.2 if (Fi > Fj) then Substitute j by the new solution; end
            6.3 Worst Solution is abandoned and replaced by new solution
            6.4 Provide ranking to the solution and find the current best solution
   End While

7. Produce the best found solution.
```

Fig. 2. Pseudo code of Hybrid Cuckoo Search algorithm.

machines, data centers, cloudlets and many other approaches for the management of different elements of the infrastructure. The main features of CloudSim are: it takes very less time or effort is for implementing the cloud computing environments. Assists modeling and simulation of large scale data centers. Supports dynamic addition of simulation elements, supports various user defined policies for allocating host to VMs, More flexible to assign different cores to more virtualized services. Permits the developers to create heterogeneous cloud environments and analyze the performance of their application services in it. Because of these extensible and generalized framework this cloudsim simulation tool is best suitable for developing customized applications.

We have tested our framework model with 151 cloud user requests and 10 cloud service providers. Here the users are requesting resources in the form of virtual machines. Let us assume that user requests for Win 7 64 bit Operating System, 2 GB RAM and 5 VM for 30 days. We have taken 1 RAM size as 512 MB and cost of it as 24, 2*GBmeansrequire*4*RAMsof* 512*MBandtheRAMcostis* 96, Single virtual machine cost is computed using

$$VM\ Cos = NO.\ of\ Days\ \times RAM\ cost \rightarrow 30 \times \$96 = \$2880$$

$$Total\ VM\ Cost = SVM\ Cost \times NO.\ of\ VMs \rightarrow \$2880 \times 5 = \$14,400$$

The user has to pay an amount of $14,400 to Cloud service provider for availing the service of 5VMs of 2GB size for 30 days. Minimum penalty cost is computed as

$$MPC = VM\ Cost \times SLA\%$$

$$MPC = \$2880 \times 25\%$$

SLA % *will differ for various CSP. Here considered the SLA% as* 25%

Out of 5 VMs if the cloud service provider is cancelling 1 VM then $720 has to be refunded as penalty along with initial cost of 1 VM. Let us assume cloud service provider has cancelled 2 VMs then penalty cost is computed using

$$Panalty\ Cost = (\text{MPC} + \text{SVM}\ Cost) \times \text{RVM}$$

$$\text{i.e}(\$720 + \$2880) \times 2 \rightarrow \$3600 \times 2 \rightarrow 7200.$$

The cloud service provider has to refund an amount of $7200 additionally $1440 with an actual amount. In case if customer needs an extension of service, No of days can be extended based on the penalty cost. No of days extended Penalty Cost / RAM Cost. i.e. $7200/$96 => 75 days for 3 VMs. i.e. (30 + 75days) for 3 VMs. Actual cost paid for cancelled virtual machines is computed using SVMC X No of reduced VMs 2880 × 2 = $5760. Cost of 2VMs for 60 days. Refunded penalty amount is $1440. i.e. 15 days. So Total No of days extended = (60 + 15) = 75 days. SLA % (percentage) is benchmarked for every Cloud Service Providers. SLA percentage differs for every CSP. It represents 3 cases:

Case 1: SLA% non-Zero: Respective CSP is providing VM to Cloud Users. CSP have cancelled some VM already bought by Cloud Users. CSP must pay Penalty cost based on SLA.

Case 2: SLA% is 0: Respective CSP is providing VM to Cloud Users. CSP is not cancelling any VM already bought by Cloud Users. CSP no need to pay Penalty cost based on SLA.

Case 3: SLA% is 0: Respective CSP is not providing any VM to Cloud Users so there is no need to calculate the penalty cost.

Fig. 3 represents SLA penalty cost computation for CSP1. Here the service provider is provisioning some kind services to the users. Service provider earned an amount of $965.89 by providing the services and at the same time, as paid a penalty of amount $139.71 for not providing the services as agreed in SLA. This represents case 1 where CSP have cancelled some VMs purchased by the user and have to pay penalty as per the SLA.

Fig. 4 represents SLA penalty cost computation for CSP 7. Here the service provider is provisioning some kind services to the users. Service provider earned an amount of $1655.82 by providing the services and not paid any penalty amount since he provides the services as agreed in SLA. This represents case 2 where CSP is providing VM to cloud users and not cancelling any VMs purchased by the user so CSP no needs to pay any penalty cost.

Fig. 5 represents SLA penalty cost computation for CSP6. Here the service provider is not all providing any kind of services to the users. So service provider has not earned any amount and doesn't to pay any penalty since he is not providing any service. This represents case 3 where CSP is not providing any VM to cloud users so there is no need to calculate the penalty cost.

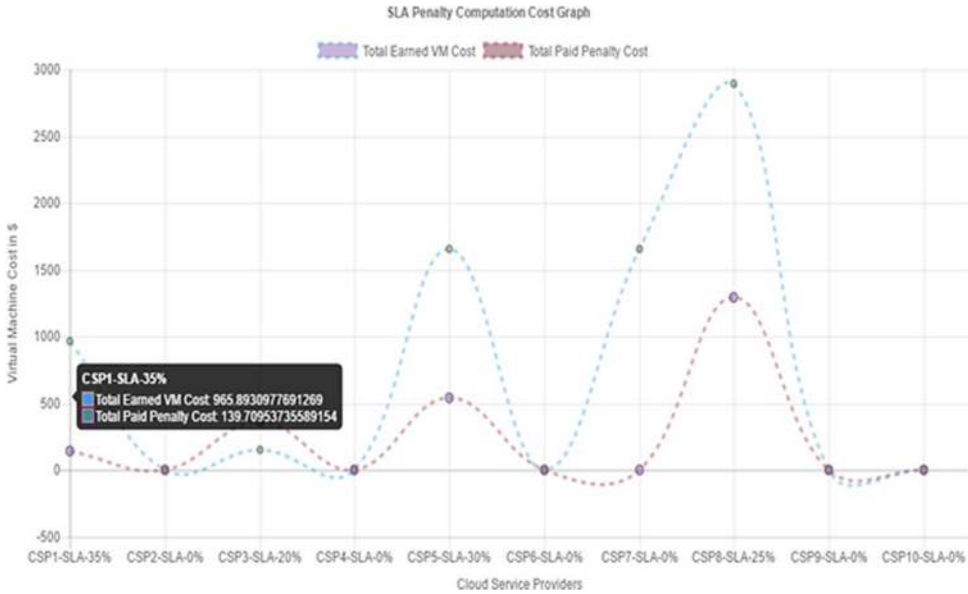

Fig. 3. SLA penalty cost computation for CSP 1.

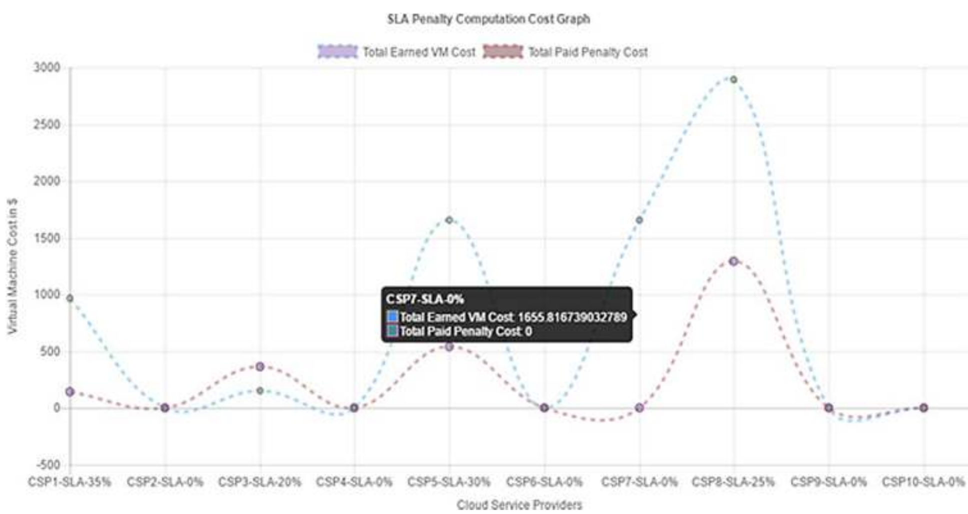

Fig. 4. SLA penalty cost computation for CSP 7.

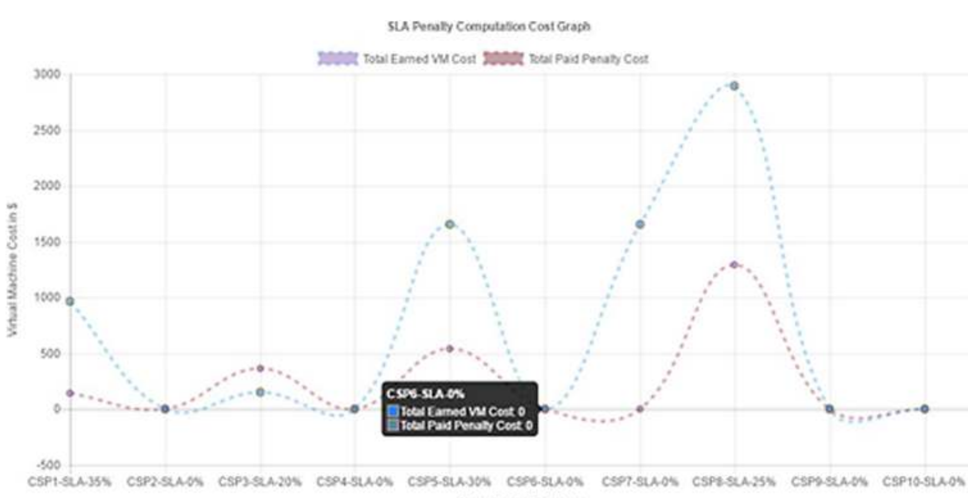

Fig. 5. SLA penalty cost computation for CSP 6.

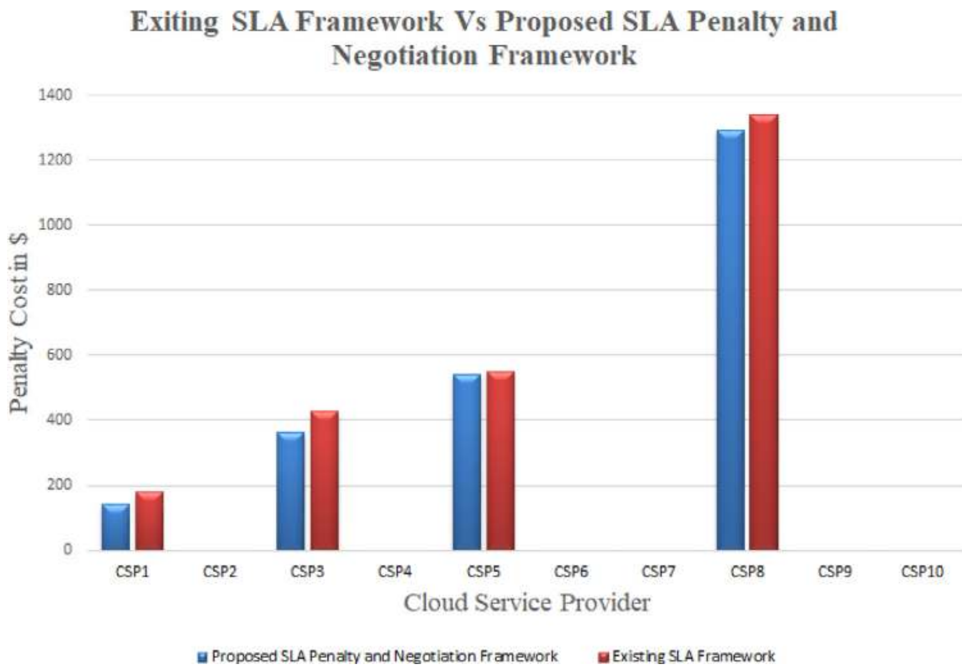

Fig. 6. Comparison of penalty cost computation using SLA model and negotiation model.

Fig. 6 represents the comparison between penalty cost computation using SLA based model and Negotiation model for all the service providers. Here, the Service providers from CSP 2, CSP 4, CSP 6, CSP 7, CSP 9 and CSP 10 offer the services without any violation.

5. Conclusion and future directions

We have designed an SLA-based resource management framework which not only take full advantage of usage of resources but as well as concentrates on different types of SLAs. This mechanism provides a guaranteed quality of service to the customers as specified in the SLA and enhance the relationship between the cloud service provider and consumer by reducing the SLA violations. The major contribution of the proposed work includes designing an Automated Resource Management Framework for Minimizing SLA Violations and Negotiation in Collaborative Cloud environment. The recommended negotiation framework primarily relies upon intellectual negotiators which accept the highlights of the autonomic computing to enrich the negotiation procedure and stay aware with the dynamic environment of the Cloud infrastructures for instance multi-tenancy, various distributed systems. Shifting over to the concept of automatic negotiation process can overcome the problems faced in the traditional static negotiation process for instance, uninterrupted changes in the commercial services requests and the nonexistence of security with the external party. Therefore, the recommended framework can minimize SLA violations and negotiation failures, and have increased cost-effectiveness. Furthermore, the recommended SLA negotiation framework is also profitable to customers subsequently customers can acquire a reasonable price reimbursement for decreased quality of service or deferring time. In future, we extend our work for ranking the cloud service providers based on their performance and Key Performance Indicators to recommend the best and suitable service provider for the cloud users. It will benefit the end users to identify the best cloud service provider based on the user requirements.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Aljoumah, E., et al. (2015). SLA in cloud computing architectures: A comprehensive study.. *International Journal of Grid Distribution Computing, 8*(5), 7–32.
Badshah, A., Ghani, A., Shamshirband, S., Aceto, G., & Pescapè, A. (2020). Performance-based service-level agreement in cloud computing to optimise penalties and revenue. *IET Communications, 14*, 1102–1112.
Buyya, R., Garg, S. K., & Calheiros, R. N. (2011). Sla-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions. In *Proceedings of the 2011 international conference on cloud and service computing* (pp. 1–10). IEEE.
de Asís López-Fuentes, F., & García-Rodríguez, G. (2016). Collaborative cloud computing based on p2p networks. In *Proceedings of the 30th international conference on advanced information networking and applications workshops (WAINA)* (pp. 209–213). IEEE..
El-Matary, D., El-Attar, N., Awad, W., & Hanafy, I. (2019). Automated negotiation framework based on intelligent agents for cloud computing. In *Proceedings of the 2019 international conference on innovative trends in computer engineering (ITCE)* (pp. 156–161). IEEE.
Foster, I., Zhao, Y., Raicu, I., Lu, S., 2008. Cloud computing and grid computing 360-degree compared. arXiv preprint arXiv:0901.0131.
Garg, S. K., Toosi, A. N., Gopalaiyengar, S. K., & Buyya, R. (2014). Sla-based virtual machine management for heterogeneous workloads in a cloud datacenter. *Journal of Network and Computer Applications, 45*, 108–120.
Hao, F., Min, G., Chen, J., Wang, F., Lin, M., Luo, C., & Yang, L. T. (2014). An optimized computational model for multi-community-cloud social collaboration. *IEEE Transactions on Services Computing, 7*, 346–358.
Hussain, A., Aleem, M., Iqbal, M. A., & Islam, M. A. (2019). Sla-ralba: cost-efficient and resource-aware load balancing algorithm for cloud computing. *The Journal of Supercomputing, 75*, 6777–6803.
Khan, S. U., & Zomaya, A. Y. (2015). *Handbook on data centers*. Springer.
Li, X., Ma, H., Zhou, F., & Yao, W. (2015). T-broker: A trust-aware service brokering scheme for multiple cloud collaborative services. *IEEE Transactions on Information Forensics and Security, 10*, 1402–1415.
Mell, P., Grance, T., et al., 2011. The NIST definition of cloud computing. Computer Security Division Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg, MD 20899-8930, pp. 1–7.
Minarolli, D., & Freisleben, B. (2014). Distributed resource allocation to virtual machines via artificial neural networks. In *Proceedings of the 22nd euromicro international conference on parallel, distributed, and network-based processing* (pp. 490–499). IEEE.
Panda, S. K., & Jana, P. K. (2017). SLA-based task scheduling algorithms for heterogeneous multi-cloud environment. *The Journal of Supercomputing, 73*, 2730–2762.
Paschke, A., & Schnappinger-Gerull, E. (2006). A categorization scheme for SLA metrics. In *Service-oriented electronic commerce* (p. eV). Gesellschaft für Informatik. Proceedings zur Konferenz im Rahmen der Multikonferenz Wirtschaftsinformatik 2006.
Peng, G. C. A., & Gala, C. (2014). Cloud ERP: a new dilemma to modern organisations? *Journal of Computer Information Systems, 54*, 22–30.
Serrano, D., Bouchenak, S., Kouki, Y., de Oliveira Jr, F. A., Ledoux, T., Lejeune, J., Sopena, J., Arantes, L., & Sens, P. (2016). SLA guarantees for cloud services. *Future Generation Computer Systems, 54*, 233–246.
Slimani, S., Hamrouni, T., & Ben Charrada, F. (2020). Service-oriented replication strategies for improving quality-of-service in cloud computing: a survey. *Cluster Computing*, 1–32.

Son, S., Kang, D. J., Huh, S. P., Kim, W. Y., & Choi, W. (2016). Adaptive trade-off strategy for bargaining-based multi-objective SLA establishment under varying cloud workload. *The Journal of Supercomputing, 72*, 1597–1622.

Suprakash, S., & Balakannan, S. (2019). Service level agreement based catalogue management and resource provisioning in cloud for optimal resource utilization. *Mobile Networks and Applications, 24*, 1853–1861.

Van, H. N., Tran, F. D., & Menaud, J. M. (2009). Sla-aware virtual resource management for cloud infrastructures. In *Proceedings of the ninth IEEE international conference on computer and information technology* (pp. 357–362). IEEE.

Wang, C., Wang, Q., Ren, K., Cao, N., & Lou, W. (2011). Toward secure and dependable storage services in cloud computing. *IEEE Transactions on Services Computing, 5*, 220–232.

Wu, L., & Buyya, R. (2012). Service level agreement (sla) in utility computing systems. In *Performance and dependability in service computing: Concepts, techniques and research directions* (pp. 1–25). IGI Global.

Zhang, W., Qian, H., Wills, C. E., & Rabinovich, M. (2010). Agile resource management in a virtualized data center. In *Proceedings of the first joint WOSP/SIPEW international conference on performance engineering* (pp. 129–140).

Zhu, Z., Bi, J., Yuan, H., & Chen, Y. (2011). Sla based dynamic virtualized resources provisioning for shared cloud data centers. In *Proceedings of the 4th IEEE International Conference on Cloud Computing* (pp. 630–637). IEEE.

K.S Sendhil Kumar works as an Associate professor in School of Computer Science and Engineering at VIT, Vellore. He received his Ph.D. in Computer science and Engineering from VIT, Vellore, ME (CSE) from Anna University and BE (CSE) from University of Madras. He has fourteen years of experience in teaching and research. He has published many papers in International and National Journals and conferences on Cloud computing, Machine Learning, IoT and Big data. He is a life member of life member of Computer Society of India. His research interests includes cloud computing, Block Chain, IoT and big data.

N. Jaisankar is Professor in the School of Computer Science and Engineering at Vellore Institute of Technology, Vellore, India. Also he was program head for MTech(CSE) programme and division head for Computer Network division. He received his BE (Computer Science and Engineering) from Bharathiar University, M.E. (Computer Science and Engineering) from M.K. University and Ph.D. (Computer Science and Engineering) from VIT University. He has 20+ years of experience in teaching and research. He received certification for CCNA Instructor and SUN certified JAVA instructor. He has reviewed many books titled Network Security, Data Mining, TCP/IP protocol suite and Programming in JAVA. He has participated as a coach in the International Programming Contest held at IIT, Kanpur, India. He has published many papers in International and National Journals and conferences on Network Security, Computer Networks and Data Mining. His research interest includes Computer Networks, Network Security, cloud computing, Data Mining. He has served in many peer reviewed International Journals as an editorial board member, Guest handling editor, advisory board member and reviewer etc. Also he has served in many international conferences as General chair, International advisory board member, technical program committee member, publication chair, organizing committee member, reviewer etc. He has worked in Neusoft Institute, Guangdong, China. He is a life member of Indian Society for Technical Education, Computer Society of India, International Association of Computer Science and Information Technology, International Society for Research in Science and Technology and member of International Association of Engineers.