Optimizing Service Level Agreements in Cloud Computing Architectures: An In-Depth Analysis

[1]M. K. Ishtiaque Ahamed, [2]Dr. Bajrang Lal

[1]Research Scholar, Department of Computer Science and Engineering, Singhania University, Pacheri Bari, Jhunjhunu, Rajasthan.

[2]Professor, Department of Computer Science and Engineering, Singhania University, Pacheri Bari, Jhunjhunu, Rajasthan

Abstract

Cloud computing services have addressed issues with Service Level Agreements (SLAs) through new research technologies. The Service-Oriented Architecture (SOA) methodology improves service quality by embracing interactions between loosely connected, reusable Web services. Parties involved are not subject to unjust penalties, and explicit specifications provide clarity. Multiple Internet Service Providers (ISPs) may be involved in a single message's delivery due to network borders. The Analytic Hierarchy Process (AHP) is used to map fuzzy logic system outcomes with CSPs, providing users with suggestions for CSPs.

Keywords: *Service Level Agreements, Cloud Computing, Architectures and Fuzzy Logic*

Introduction

The performance guarantee that the customer and cloud service provider establish is called a Service Level Agreement (SLA). In the past, while using cloud computing, every SLA was worked out directly between the customer and the service provider. Most Service Level Agreements are standardised these days until a client becomes a big user of cloud services, thanks to the advent of big utility-like cloud computing providers.

The terms and conditions under which a service provider agrees to supply its services to a customer are laid out in a legally binding document known as a service level agreement (SLA). Consequently, the SLA is a collection of requirements that must be met in order for services to continue. The document guarantees the customer a certain level of service quality (QoS) and outlines the procedures for penalties in the event of service conditions being violated. No technology, including cloud computing, can hope to achieve its stated goal of "Cheaper, Faster and Better" technology delivery. Cloud computing has almost achieved the first two goals, and in order to accomplish the third, it aims to help information technology (IT) provide as many services as possible to customers, whether they are businesses or individuals, who want to utilise IT as a service.

A better way to acquire cloud services is to use monitoring services in a way that makes cloud providers as much money as possible while cloud users pay as little as possible. In order to successfully monitor the services supplied, a well-formatted and maintained SLA is required. This is because the SLA details every important and deciding aspect of the cloud service. Web-Service Level Agreement (WSLA), Web-Service Offering Language (WSOL), and SLAng are some of the well-established SLA frameworks for Web-Services. The main ways in which these frameworks differ from one another are in the language used for implementation, the parameters that are taken into account, the way constraints are imposed, and the life cycle that is adopted.

Web-services and cloud computing have many similarities in their service level agreements (SLAs) due to their shared use of service-oriented architecture. However, there are a number of other factors to think about since there is a wide variety of Cloud computing service deployment types (e.g., SaaS, PaaS, and IaaS). Finding a suitable SLA framework for the cloud is challenging since current frameworks are either web-service specialised, still under development, or missing key capabilities like SLA negotiation. Furthermore, the majority of SLA frameworks used by webservices are static. The variety of Cloud apps makes it impossible for the current static

SLA templates to handle them. Furthermore, the majority of frameworks only account for cost and performance assessments, ignoring crucial non-functional criteria like trust, dependability, interoperability, risk, and green computing.

Literature Review

Bajrami, Enes et.al. (2022). An SLA is a legally binding agreement between a customer and its cloud provider that specifies the expected level of service. Since large utility-like cloud computing providers have emerged, most SLAs are now typical until a customer becomes a heavy user of cloud services. However, as per the agreement, the issuer of a cloud service must compensate the cloud provider client for any failure to meet the specified minimum goals. Therefore, SLAs are similar to insurance policies in that the business is liable for paying the amount specified in the agreement in the event of a loss. In order to determine if businesses in North Macedonia employ SLAs when purchasing or selling, we have developed a questionnaire. The purpose of this article was to lay down the groundwork for a survey that would determine if businesses in North Macedonia are making use of SLA in their agreements with one another.

Lo Piparo, Teo. (2020). Service Level Agreement (SLA) administration becomes more complicated with the ever-changing nature of cloud computing, which affects both service requirements and pricing structures. The ineffectiveness of service level agreements un cloud computing, and more specifically in IT outsourcing, is the target of this research. Interviews with experts in the field and those working in the field will provide the backbone of the study approach for understanding dynamic SLA management. In order to track the SLA lifecycle's dynamic iterations, the research polled five companies in Ireland and Western Europe. Researchers uncovered the "ex habitus" dynamic phenomena, which improved iteration planning and made risk aversion a top priority for practitioners. The results provide some relief from the issue at hand by expanding our understanding of SLA life-cycle management in the cloud. In order to help people better understand and rationalise SLA agreements, we provide a model for cloud computing SLA negotiation.

de Azevedo, Leonildo et.al. (2018). Cloud computing offers on-demand, transparent services to clients, but fulfilling Service Level Agreements (SLAs) remains a challenge due to service demand and system configuration. Load balancing and resource provisioning algorithms are necessary for efficient resource distribution. However, existing studies do not effectively address resource provisioning problems using optimization techniques, limiting the analysis to a limited set of objectives. This paper proposes algorithms to address computational resource provisioning problems using optimization techniques on-the-fly, optimizing the use of resources in the cloud infrastructure to fulfill client requirements and ensure efficient resource use. This paper aims to address the computational resource provisioning problem effectively.

Al Kim, Haider et.al. (2018). Cloud computing offers future-proof IT infrastructure, adaptability, and service quality. Its modern systems, such as cluster and grid computing, aim to achieve maximum output with minimal effort or resources. However, concerns about privacy, security, and service quality have grown with technological progress. Fast service delivery for commercial apps is crucial, and the Service Level Agreement (SLA) is necessary to accommodate the trend of pay-as-you-go for most corporate apps. Cloud computing differs from traditional computer systems and requires a format and criteria for evaluating SLAs to ensure both customers and providers get the most out of it.

Radha, Karampudi et.al (2015). Present day massive amounts of data are present in most sectors. The range of data sizes is Tera bytes to Peta bytes. Data expansion is something that organisations are trying to manage. Businesses are tackling big data and analytics in relation to the cloud's interaction with big data via installations in the cloud. This article lays out the current state of large data processing in distributed contexts, as well as future research prospects in this area. The Institute of Advanced Engineering and Science has the copyright for the year 2015. This is a copyright notice.

Research Methodology

Concepts of customer-based SLA on SAAS billing are the main emphasis of the suggested technique. Figure 1 show the proposed architecture, which aims for a SAAS billing model based on subscription fees and a customer-based SLA. Users of cloud services contact the Intelligent Third Party, which is chosen from among the subscribers, to provide the services they need. The ITP uses Fuzzy Logics to record the procedures and requirements. Consequently, the criteria are specific

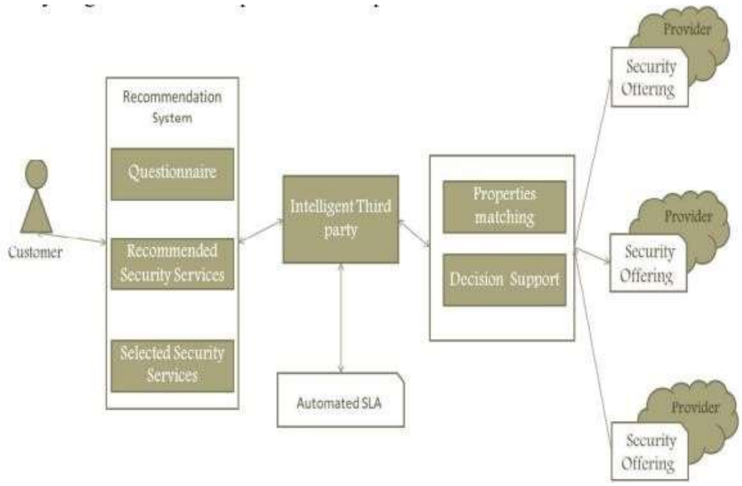

Fig.1 Proposed Architecture

Data about the service offers of different cloud providers, together with ratings on specific elements like scope, quality, and responsibility, are included in the ITP. The Analytic Hierarchy Process (AHP), a Decision Support System (DSS), determines which Cloud Service Provider is most suitable for the Cloud Service User based on this data of CSPs and the exact criteria of CSU. If the CSU is dissatisfied with the outcome, the recommendation is made for the next best CSP. An SLA outlining the service's specifics and the needs of its consumers is created when the CSU is fulfilled. We have sent this SLA to the CSP since it is not yet complete. In order to subscribe to the services, the CSU contacts the CSP. There are several options available for each service. Once the proposal is approved, a comprehensive SLA will be created by both the cloud service provider and the cloud service user.

Results And Analysis

Envision User-X, a customer of cloud services, contacting an Intelligent Third Party (ITP) to choose and subscribe to services via a cloud service provider (CSP). The ITP has registered SP-A, SP-B, and SP-C as CSPs. User-X chooses a service and fills out an ITP survey. Fuzzy logic is used to process requirements that are obtained from the questionnaire. The AHP is fed the outcomes via these procedures. The result is the most appropriate cloud service provider for the service that User-X selected. It uses an agent simulation to verify our negotiating model based on brokers. It features three distinct time-based decision-making roles—the polynomial, the sigmoid function, and the exponential—and has used Java with JRE 1.4 to improve the database system software and IBM DB2.

In order to choose a suitable negotiation plan, the Intelligent strategy selection algorithm suggests a mathematical mapping model, and the Negotiation Broker uses this model to generate the strategy's parameters. The parameters of the negotiating tactic might be updated using the adaptive algorithm (Zulkernine and Martin, 2011). The plan made it very apparent how the service recipients would provide the functional and nonfunctional needs for different services. Nevertheless, not all clients will be able to provide the exact specifications that are suitable for their company because of its rigidity or difficulty in accommodating their needs.

Selecting Services

A number of service providers' SAAS offerings are included below, along with the services themselves and the names of those providers. You may narrow down the services by service type or by keyword. Fast Presentations and Creating UML are two services that come to mind. Along with customer reviews, every service also comes with screenshots. The ITP displays a list of services; UserX chooses the ones they need. Suppose User X has chosen Quick Presentations as the service in our example.

Collecting Requirements using Fuzzy Logic

Fuzzy logic will be used to gather questionnaires from clients. The characteristic "customizability" is shown below as an example; Adaptability upon request:

Q1. In your opinion, how well do the service's screenshots portray its overall design? Within the range of 0 to 100%

Q2. What number of service UI levels (fonts, icons) would you want to be able to customise? Within the range of 0 to 100%

The quantity of linguistic variables is equal to the number of attribute inquiries. Two linguistic factors, X (satisfaction with the overall appearance) and Y (changes to the user interface), may have values of 35% and 60%, respectively. For X, the sets are High, Medium, Low, and for Y, it's Small, Large.

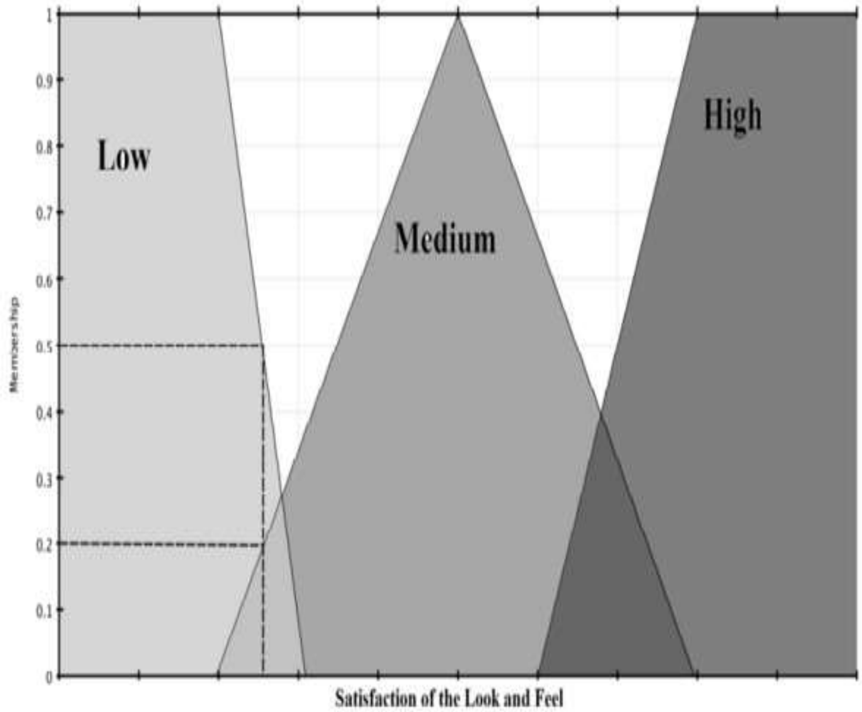

Fig. 2 Visual Representation of Satisfaction of Look and Feel

$$\mu satisfaction(X)=low(35)=0.5$$

$$\mu satisfaction(X)=medium(35)=0.2$$

$$\mu satisfaction(X)=high(35)=0.0$$

$$\mu change(Y)=small(60)=0.1$$

$$\mu change(Y)=large(60)=0.7$$

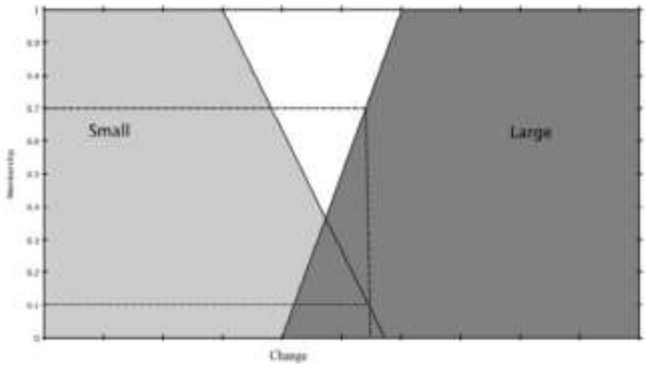

Fig. 3 Visual Representation of Change

From trapezoid fuzzy sets, the membership functions are produced. The boundary values of a trapezoid fuzzy set are 4. Going from left to right, middle to right.

- If value <= left, μvalue = 0.
- If left <= value < left_middle, μvalue = (value - left) / (left_middle – left).
- If left_middle <= value < right_middle, μvalue = 1.
- If right_middle<=value < right, μvalue = (right – value) / (right – right_middle).
- If right <= value, μvalue = 0.

The Fuzzy Rules

When X is little and Y is big, personalisation is highly valued. Strongly favoured is customizability when X is Medium. Modest preference for customisation is indicated by low X and small Y. Assuming X and Y are both high, the preference for customizability is low. We do not recommend customisation if X is large and Y is little.

μcustomizability=Extreme=min[μx(a),μy(a)]=min[0.5,0.7]=0.7

μcustomizability=Strong= μx(a) = 0.2

μcustomizability=Moderate=min[μx(a),μy(a)]=min[0.5,0.1]=0.1

μcustomizability=Light=min[μx(a),μy(a)]=min[0.0,0.7]=0.0

μcustomizability=No = min[μx(a), μy(a)] = min[0.0,0.1] = 0.0

Based on the information provided, Customizability is highly valued and ranked as 5 in Table 2. As a result, User-X chooses a score for each service characteristic that represents their preference level. The formula for determining customisation is given in Eqn. 4.1. The following method aids service recipients by meeting their needs for requirements with additional value. Here, we use the "Boolean logical" values concept to set the service attribute's truth values to 0 or 1.

Thus, the variables' truth values will be either 0 or 1, as opposed to the actual numbers (0–1) used in fuzzy logic systems. Fuzzification and Defuzzification are membership functions that deal with the linguistic units of these variables. All the different kinds of service qualities are considered. Van Broekhoven and De Baets (2006) assessed the performance of a fuzzy logic approach that relies on membership functions. Defuzzification may be accomplished using a variety of methods. Most people use the centroid approach (Greenfield et al., 2009).

$$\text{Result} = \frac{((30+40)*0.1)+((50+60)*0.2)+((70+80+90+100)*0.7)}{(0.1*2)+(0.2*2)+(0.7*4)}$$

Customizability Preference value = 78.5%

All of the other properties listed in Table 1 follow the same procedure. Equation 4.1 is used to determine the preference value of the other qualities. Almost all customers and users prefer a rating of 5, which indicates a level of efficiency that is 99 percent of the time evaluated by the consumer. Similarly, service response time is the most significant service characteristic that consumers desire and anticipate, ranking fifth. A lack of universal desire for data privacy is the reason it is rated as '1'. Based on the literature review (O'Brien, L et al., 2007) and our own thoughts (Table 1), this crucial part takes into account all of the chosen QoS criteria. Given a set of qualities, what follows are the values that people would like

Table 1 Preference Table for Attributes

Attribute	Preference Value	Preference Range	Preference Ranking
Customizability	78.5%	Extremely Preferred	5
Performance	60%	Strongly willing to perform	4
Efficiency	99%	Willing to achieve Extreme Efficiency	5
Suitability	75%	Strongly Preferred	4
Low Cost	65%	Strongly Preferred	4
Availability	99%	Extremely Preferred	5
Reliability	70%	Strongly Preferred	4
Service Response Time	88%	Extremely Preferred	5
Scalability	54%	Moderately Preferred	3
Efficiency (of provider)	70%	Strongly Preferred	4
Performance	90%	Extremely Preferred	5
Data Integrity	99%	Extremely Preferred	5
Data Privacy	10%	Not Preferred	1
Maintainability	30%	Lightly Preferred	2

Service Offering by Various CSP's

Table 2 displays the ranking of preferences based on the services offered by various cloud service providers, as indicated during the service composition. All registered and privileged consumers in the cloud were given access to these offers so that they could evaluate and sign up for services according to their needs. Such tabular information are very helpful for cloud service providers, users, and clients. In order to provide consumers with the finest recommendations, Intelligent Third Party (ITP) use this process. Cloud Service Users (CSUs) may find this offering analysis useful in evaluating all available Cloud Service Providers (CSPs) and selecting the one that best suits their needs. By comparing the services provided by different service owners or providers, this analysis ranks the qualities (see Table 3) and ultimately helps the user pick the best service.

Table 2 Ranking Preference Table

Preference	Ranking
Not Offered	1
Lightly Suitable	2
Moderate Efficiency	3
Strongly Offered	4
Extremely Offered	5

The list of service offers is provided by different CSPs. For instance, CSP - A's offers

---

Table 3 Preference Table for CSP-A

Attribute	Offered Value	Preference Range	Preference Ranking
Customizability	10%	Not Offered	1
Performance (of users)	70% (Average performance of other users)	Strongly Performed	4
Efficiency (of users)	50% (Average efficiency of other users)	Moderate Efficiency	3
Suitability	30% (Average user feedback rating)	Lightly Suitable	2
Low Cost	85%	Extremely Offered	5
Availability	90%	Extremely Offered	5
Reliability	60%	Strongly Offered	4
Service Response Time	54%	Moderately Offered	3
Scalability	35%	Lightly Offered	2
Efficiency	95%	Extremely Offered	5
Performance	52%	Moderately Offered	3
Data Integrity	50%	Moderately Offered	3
Data Privacy	30%	Lightly Offered	2
Maintainability	80%	Extremely Offered	5

The same holds true for the service providers CSA-B and CSA-3; their offers are computed.

Analytic Hierarchy Process

The rating for the services is provided in Table 3, which tabulates the preference ranking. Finding solutions to composite issues with several criteria is what the Analytical Hierarchy Process (AHP) is all about. The goal of AHP is to streamline variety creation by establishing priority among many replacements and criteria. As part of the process's design, the data manager will decide how each standard is relative to the others and will then speculate on which criteria to use when replacing resolutions. The result of using AHP is a priority ranking that shows how each choice option is rated in terms of overall preference. The criteria or features indicated in Table 5 range from sustainability in Table 4 (A1 to A14) to customizability.

AHP Matrix

Alternative 1	A1/A1	A1/A2	..	A1/An
Alternative 2	A2/A1	A2/A2	..	A2/An
...	...	...	..	...
Alternative n	An/A1	An/A2	..	An/An

The analytic table was constructed using the preference table and the qualities mentioned in Table 4, as shown by the above AHP Matrix. The ITP-affiliated providers will showcase the security solutions they provide. ITP will keep track of information on the security offerings of each cloud service provider in a database. Here is how the information on each CSP is kept:

Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

- ID of the CSP
- the name of the CSP
- A catalogue of the security services offered
- Most desirable raking rates for the fourteen criteria we covered before.

Every CSP's security service catalogue is cross-referenced with the client's chosen security criteria. Unfortunately, not every CSP is able to meet every client's specific security needs. Therefore, for every CSP, the Match % is determined. Under matched CSPs, the CSP is chosen if the match percentage is greater than or equal to 70%. Here, either all of the CSPs meet the client's security needs or none of them do. After seeing the matching number of CSPs, the client may decide to modify the criteria. So, we provide the customer the opportunity to return and adjust the security settings as needed.

Table 4 Analytic Table Based on Preference Table

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	RP
A1	1	5/4	1	5/4	5/4	1	5/4	1	5/3	5/4	1	1	5	5/2	0.089
A2	4/5	1	4/5	1	1	4/5	1	4/5	4/3	1	4/5	4/5	4	2	0.07
A3	1	5/4	1	5/4	5/4	1	5/4	1	5/3	5/4	1	1	5	5/2	0.089
A4	4/5	1	4/5	1	1	4/5	1	4/5	4/3	1	4/5	4/5	4	2	0.07
A5	4/5	1	4/5	1	1	4/5	1	4/5	4/3	1	4/5	4/5	4	2	0.07
A6	1	5/4	1	5/4	5/4	1	5/4	1	5/3	5/4	1	1	5	5/2	0.089
A7	4/5	1	4/5	1	1	4/5	1	4/5	4/3	1	4/5	4/5	4	2	0.07
A8	1	5/4	1	5/4	5/4	1	5/4	1	5/3	5/4	1	1	5	5/2	0.089
A9	3/5	3/4	3/5	3/4	¾	3/5	3/4	3/5	1	3/4	3/5	3/5	3	3/2	0.053
A10	4/5	1	4/5	1	1	4/5	1	4/5	4/3	1	4/5	4/5	4	2	0.07
A11	1	5/4	1	5/4	5/4	1	5/4	1	5/3	5/4	1	1	5	5/2	0.089
A12	1	5/4	1	5/4	5/4	1	5/4	1	5/3	5/4	1	1	5	5/2	0.089
A13	1/5	1/4	1/5	1/4	1/4	1/5	1/4	1/5	1/3	1/4	1/5	1/5	1	1/2	0.018
A14	2/5	2/4	2/5	2/4	2/4	2/5	2/4	2/5	2/3	2/4	2/5	2/5	2	1	0.035
VS	11.2	14	11.2	14	14	11.2	14	11.2	18.7	14	11.2	11.2	56	28	1

In order to meet the security criteria, a basic matching algorithm is developed. An array is used to hold the unique identifiers for each security requirement. Every CSP's match count is raised by one if one of their security service IDs is in the previously stored list of IDs.

Relative Priority Calculation

Below, in Table 5, you can see the calculated results of the relative priorities of the service providers with regard to each characteristic.

Table 5 Relative Priority Respect to Customizability

Customizability	SP-A	SP-B	SP-C	Relative Priority
SP-A	1	1/5	1/3	0.111
SP-B	5	1	5/3	0.556
SP-C	3	3/5	1	0.333
Vertical Sum	9	1.8	3	1

Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

One of the most crucial service criteria or qualities for achieving software quality attributes like flexibility and dependability is customisation. Table 6 shows the RP for the service provider's supplied services according to the "Customizability" criterion.

Table 6 Relative Priority Respect to User Performance

Performance (of a user)	SP-A	SP-B	SP-C	Relative Priority (RP)
SP-A	1	1	4/3	0.364
SP-B	1	1	4/3	0.364
SP-C	¾	3/4	1	0.272
Vertical Sum	2.75	2.75	3.667	1

Table 7 displays the relative priority (RP) metrics, which are derived from customer performance on the part of a certain service provider. This performance metric shows how efficiently the service provider's customers and clients used the service. In order to determine the optimal service composition, efficiency is a crucial quality feature to assess. For the "Efficiency" characteristic, the RP is in Table 8. In a similar vein, we will determine the Relative Priority for each characteristic.

Table 7 Relative Priority Respect to User Efficiency

Efficiency (of a user)	SP-A	SP-B	SP-C	Relative Priority
SP-A	1	3/5	3/4	0.25
SP-B	5/3	1	5/4	0.416
SP-C	4/3	4/5	1	0.333
Vertical Sum	4	2.4	3	1

Calculation of each service provider's weight values (SP)

An organization's service provider's weight

$$P=\sum_{i=1}^{n} r_{ci} * r_{ai}$$

Where, r_{ci}, stands for an attribute's proportional importance.

r_{ai}, Shows how important that characteristic is to the service supplier.

Weight value of SP-A = (0.089*0.111) + (0.07*0.364) + (0.089*0.25) + (0.07*0.1818) + (0.07*0.385) + (0.089*0.333) + (0.07*0.307) + (0.089*0.23) + (0.053*0.2) + (0.07*0.357) + (0.089*0.23) + (0.089*0.23) + (0.018*0.166) + (0.035*0.333) = 0.259

Weight value of SP-B = (0.089*0.556) + (0.07*0.364) + (0.089*0.416) + (0.07*0.454545) + (0.07*0.229) + (0.089*0.333) + (0.07*0.307) + (0.089*0.384) + (0.053*0.4) + (0.07*0.285) + (0.089*0.384) + (0.089*0.384) + (0.018*0.416) + (0.035*0.333) = 0.3738

Weight value of SP-C = (0.089*0.333) + (0.07*0.272) + (0.089*0.333) + (0.07*0.363636) + (0.07*0.385) + (0.089*0.333) + (0.07*0.384) + (0.089*0.384) + (0.053*0.4) + (0.07*0.357) + (0.089*0.384) + (0.089*0.384) + (0.018*0.416) + (0.035*0.333) = 0.355

The formula for determining the weightage is given in Eqn. 4.2. Accordingly, SP-B is the most heavily weighted. Therefore, User-X should go with SP-B as their Cloud Service Provider.

Conclusions

Manually describing service needs is not very successful in these new cloud features, according to all the obvious research of current SLA solutions. In order to match CSPs' security offers with cloud customers' security needs, the negotiation mechanism takes these requirements as inputs. The Decision Support System's AHP model is used to identify the most appropriate CSP from the pool of matching CSPs. In the end, the customer's interest determines the generation of a SLA. It is not within the purview of this study to monitor the SLA phase; such monitoring may be part of an additional project.

References

[1] A. Vijaya and N. Venkataraman, "A model driven framework for portable cloud services," International Journal of Electrical and Computer Engineering (IJECE), vol. 6, pp. 708–716, 04 2016.

[2] Al Kim, Haider & Barua, Shouman. (2018). Service Level Agreement (SLA) for Cloud Computing Compilation with Common and New Formats. International Journal of Scientific Research and Management. 6. 10.18535/ijsrm/v6i4.ec01.

[3] Amazon, "Amazon EC2 spot cloud," http://spotcloud.com/, 2016, [Online; accessed 18-January-2016].

[4] B. Javed, P. Bloodsworth, R. U. Rasool, K. Munir, and O. Rana, "Cloud Market Maker: An automated dynamic pricing marketplace for cloud users," Future Generation Computer Systems, vol. 54, pp. 52 – 67, 2016.

[5] Bajrami, Enes & Halili, Festim. (2022). Service level agreement for cloud computing and usability in public and private enterprises in north macedonia.

[6] D. Technologies, "6 cloud computing trends for 2018," Nasscom Community, 2018.

[7] de Azevedo, Leonildo & Estrella, Júlio & Nakamura, Luis H. V. & Santana, Marcos & Santana, Richard & Toledo, Claudio & Batista, Bruno & Reiff-Marganiec, Stephan. (2018). Optimized Service Level Agreement Establishment in Cloud Computing. Computer Journal. 61. 1429-1442. 10.1093/comjnl/bxx087.

[8] J. Sahni and D. P. Vidyarthi, "A cost-effective deadline-constrained dynamic scheduling algorithm for scientific workflows in a cloud environment," IEEE Transactions on Cloud Computing, vol. 6, no. 1, pp. 2–18, 2018.

[9] K. M. Sim, "Agent-based cloud computing," IEEE Transactions on Services Computing, vol. 5, no. 4, pp. 564–577, Fourth 2015.

[10] Lo Piparo, Teo. (2020). The challenges with the service level agreement for the cloud computing buying organization A negotiation guideline for ever-changing contract requirements based on transaction cost economics. 10.13140/RG.2.2.12269.41443.

[11] Mirobi, G. & Arockiam, L.. (2015). Service Level Agreement in cloud computing: An overview. 753-758. 10.1109/ICCICCT.2015.7475380.

[12] Odun-Ayo, Isaac & Udemezue, Blessing & Kilanko, Abiodun. (2019). Cloud Service Level Agreements and Resource Management. Advances in Science, Technology and Engineering Systems Journal. 4. 228236. 10.25046/aj040230.

[13] PlanForCloud, "Planforcloud: Cloud portfolio management," https:// www.planforcloud.com/, 2015, [Online; accessed 03-November-2015].

[14] Radha, Karampudi & Rao, Dr. B.Thirumala & Babu, Shaik & Komati, Thirupathi Rao & Reddy, Vuyyuru & Saikiran, P.. (2015). Service Level Agreements in Cloud Computing and Big Data. International Journal of Electrical and Computer Engineering (IJECE). 5. 158-165. 10.11591/ijece.v5i1.pp158-165.

[15] S. Singh, I. Chana, and R. Buyya, "STAR: SLA-aware autonomic management of cloud resources," IEEE Transactions on Cloud Computing, 2017.