Novel Database Driven Architecture (DDA) for QoS Maintenance and Management in Cloud Computing

S.Beghin Bose M,Sc(SE),Reg No:17213152271004 Full time Ph.D scholar in computer science, ST Hindu college, Nagercoil,India, Manonmaniam Sundaranar University, Abishekapatti,Trinelveli,627012,Tamilnadu,India, beghinbose.s@gmail.com

Dr.S.S.Sujatha MCA, M.phil, PhD, Associate Professor, ST Hindu College Nagercoil, India, sujaajai@gmail.com.

Abstract - The Cloud service has become a vital component of modern information systems where the quality of service (QoS) has a direct influence on the system performance. It has been possible due to the rapid development of cloud computing. Cloud services are offered by Cloud Service Providers (CSPs) to the cloud customers in the pay-per-use model and the service providers use Service Level Agreements (SLAs) in order to define the quality of the services offered. SLAs are just contracts which characterize the performance and quality of the services offered by CSPs. QoS values of cloud services provide valuable information to assist decision making while making optimal cloud service selection from a set of functionally equivalent services. A perfect QoS model has not still been proposed by anyone yet to the best of our knowledge [6][7][8]. In this paper a novel Database Driven QoS architecture is introduced in order to increase the trust of cloud service user on CSPs.

Keywords —Cloud Applications, Cloud Services, Optimal Service Selection, Quality-of-Service, QOS Management, QOS Monitoring.

I. INTRODUCTION

Cloud computing is nothing but Internet-based computing, in which shared configurable resources like infrastructure, platform, and software are provided as services to computers and other devices. Due to the popularity of Cloud Services, many leading IT firms including Google, IBM, Microsoft, and Amazon have started to offer cloud services to their customers. Large scale and complex applications have been deployed in the cloud environment. Similar to traditional component-based systems, cloud applications typically involve multiple cloud components communicating with each other over application programming interfaces, such as through web services. In the traditional component base systems the software components are invoked locally but in the cloud applications cloud services are invoked remotely by internet connections. As there are umpteen functionally equivalent services in the cloud, optimal service selection becomes vital. A parameter called QoS is presently used to measure the nonfunctional performance of cloud services. As the popularity of cloud computing rises rapidly, how to build high quality cloud applications becomes an urgently-required research problem.

Private, community, public, and hybrid clouds are generally regarded as deployment models for clouds. According to their requirements users can choose one of these services and deployment models. Irrespective of the advantages of using clouds, there are a few obstacles and concerns. One such concern is getting QoS guaranteed by service providers. In clouds, user applications with different characteristics and demands are executed. In order to define the requirements of each application, a contract called SLA is negotiated between users and cloud service providers. According to this contract, a provider is penalized when the provider fails to meet the mutually agreed SLA, There are vital challenges in modifying traditional algorithms to satisfy SLAs properly so much so designing architectures and algorithms for managing SLAs in clouds is still in its early stages.

There are two ways to measure QoS of cloud services. It can be measured either at the client side (e.g., response time, throughput, etc.) or at the server side. Performance of cloud services at the client side is greatly influenced by the unpredictable Internet connections. Hence, how to help the users to select the cloud service that can most satisfy user's needs is one of the important problems. Thus in general, selection of a service from the candidate set is determined by QoS metrics such as response time, reputation, and reliability. Our proposed architecture consists of three modules: metric monitoring, database and analyzer.

- Metric Monitoring: It is monitoring the cloud metrics of cloud service providers using available technologies.

 DOI : 10.18231/2454-9150.2018.1111

- Database: Monitored metric information is stored in database.
- Analyzer: Analyzer checks the monitored information as well as whether the services promised are fulfilled.

1.1 QoS IN CLOUD COMPUTING

When a company, organization or an individual decides to buy a cloud SLA is of paramount importance [1]. SLA is a contract negotiated between a user and service provider; it consists of minimum level of information regarding the services offered. SLA varies according to the service provider. SLA becomes vital when the services, applications and data of an organization are changed over into cloud. In this scenario, SLA should include all the enterprises level requirements. Technical definitions are generally used in SLA that qualify the level of service, such as mean time between failures. SLA must include a few more conditions: disaster recovery information which means how data is recovered during a disaster; security and data privacy information, which consists of information regarding how data is secured in cloud. Details of compensation, if the provider fails to satisfy the user should also be mentioned in SLA. At last the SLA should include an exit strategy which ensures a smooth transition. Only after an SLA is agreed between a service provider and a user the cloud service will be implemented for the particular organization or individual. Our proposed architecture consists of three modules: metric monitoring, database and analyzer.

1.2 QoS METRIC

QoS Metric is classified into performance, dependability and configuration [2]. Performance indicates the quality of a service offered by a service provider. Fig.1 shows the tree level structure of cloud metrics.

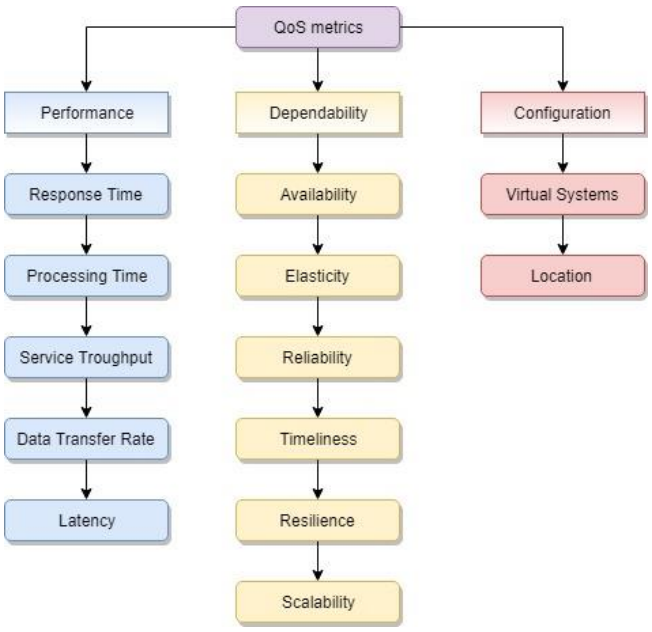

Fig.1 Cloud metrics details

Performance is divided into response time, processing time, service throughput, data transfer rate and latency. Response time is the time taken by a cloud provider to give a response for a request made by a cloud user. Response time is very important as it gives an overall picture of cloud's performance. According modern education and computer science publisher, average overall response time is 50.05 milliseconds if the data centers are placed at the same location. There is significant increase in response time if the data centers are placed in different locations i.e. 401.7 milliseconds. Throughput is a unit of work processed by cloud in a unit of time. It varies according to the work load of different cloud users. It is normally measured as transaction per second whereas in bulk data processing it is measured as data rate. It is vital to have enough throughput in order to run all applications with optimal efficiency. Data transfer rate measures how fast the service is provided. Latency means the time interval between submitting a packet and it reaching the destination [3][4]. As latency increases cloud performance decreases.

Dependability includes availability, reliability, elasticity, timeliness, resilience, scalability, and security. Availability is nothing but how much time cloud is available in a day or in a month or in a year. According to a study conducted by IWGCR, cloud is unavailable for only 7.738 hours in a year that means cloud is available for 99.91 percentage of the time. Hence there is 42 minutes of down time per month during which cloud service is not available. Elasticity refers to clouds' ability to expand and contract overtime in response to user demands. All electronic products are prone to malfunction. Reliability means how fast a malfunctioning in cloud is recovered. It is defined by two parameters: one is mean time between failures and another one is mean time to repair. The average amount of time a device functions before failing is called mean time between failures. Mean time repair means the average time taken to fix the failed component and the device is returned to the production status. Managing any number of requests received from users at a time is called scalability. A better scalable cloud will satisfy any number of requests from users at a time without failure. Scalability is vital to evaluate as it determines whether a system can handle a huge number of request applications simultaneously. Security is a very vital metric as confidential matters are stored in cloud. A set of control-based technologies and policies devised to protect data applications; important information and infrastructure related to the use of cloud computing can be described as security. According to Skyhigh security risk report 86 percentage organizations face at least one security threat per quarter.

Configuration includes several metrics indicating the configuration status. It is the process of setting hardware and software details for components of a cloud to ensure that they can interoperate and communicate.

 DOI : 10.18231/2454-9150.2018.1111

The remainder of this paper is organized as follows: Proposed model presented in Section II; Results and discussion in Section III; Performance analysis in section IV Conclusion in section V.

II. PROPOSED MODEL

Our proposed model involves three important steps shown in fig2. The first step is a metric monitoring of the CSP. After the monitoring process, the measured values are stored in a database. Finally the analyzer checks the monitored information and agreed QOS information.

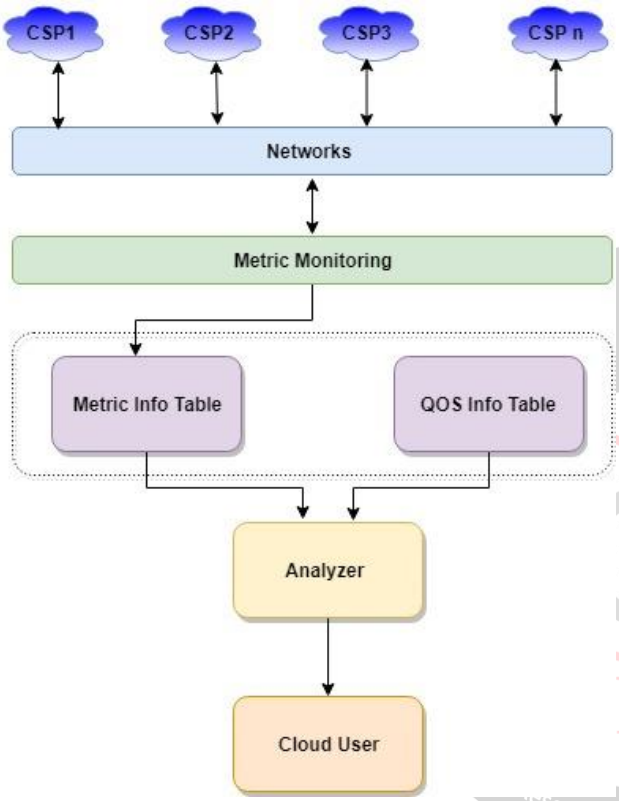

Fig.2 Components of the DDA architecture

2.1 METRIC MONITORING

Of all the metrics which have been discussed above, now we know which the most important ones are. We have to find out how to measure the metrics. IT Department and Cloud Auditors have been using already developed tools to measure metrics. In our proposed model we choose PerfKitBenchmarker to measure the cloud metrics [5]. PerfKit Benchmarker is licensed under the Apache 2 license terms. PerfKit Benchmarker is a community effort involving over 500 participants including researchers, academic institutions and companies together with the originator, Google

2.2 DATA BASE

In order to achieve efficiency and accuracy in measurements a database driven design is included in our proposed DDA architecture. We use two main tables in the proposed system: Metric Info Table (MIT) and QoS Info Table (QIT). The MIT is used to store important performance metric information. SCP ID, User ID, Date & Time, response time, processing time, data transfer rate and latency are the information stored in the MIT.

- CSP ID: Every service provider has a unique ID.
- User ID: Every cloud user has a unique ID.
- Date & Time: Log in date and time of the cloud user.
- Response Time: The time taken by the cloud service provider to give response to the cloud user. It is measured in millisecond.
- Processing Time: Time taken to complete a process. It is also measured in millisecond.
- Data Transfer Rate: Rate of speed in which data is transferred. It is measured in mbps.
- Latency: The time taken for a data packet to be sent from a source to a destination. It is measured millisecond.

All other metrics are also measured and stored in the Metric info database in the same manner. When a user purchases cloud service; the CSP gives QoS details. These details are stored in QIT. The main attributes in the QIT are CSP ID, Cloud Service ID, Cloud User ID, Purchasing Date of the Service, Date of expiry of the contract and also the details of the above mentioned metrics. The details are stored when contract is signed between cloud service user and cloud service provider. The details stored in MIT and the details stored in QIT are maintained as historical information.

2.3 ANALYZER

This is the final stage of our architecture. Analyzer checks MIT and QIT to find out whether the promised QoS is fulfilled by the cloud service provider. Cloud user can directly communicate with the analyzer. Analyzer gives a detailed report of QoS when a user signs out from cloud. CSU can view this report any time. Hence the CSU's trust on QoS multiplies.

2.4 Monitoring and SLA management

Monitoring and SLA management is performed by the proposed framework, whose architecture is shown in fig 6.

 DOI : 10.18231/2454-9150.2018.1111

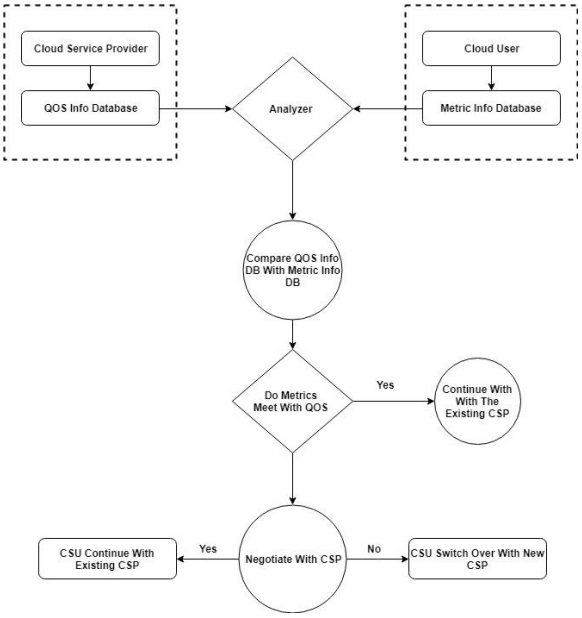

Fig.3 Process flow of proposed work

Step 1: Agreed SLAs are stored in QOS info table.
Step 2: Monitored metrics are stored in Metric Info Table.
Step 3: The analyzer monitor the Metric Info Table and QOS info table within a particular time interval.
Step 4: If the measured metric information is not satisfied with agreed QOS, it sends alert to the cloud user.
Step 5: The CSU negotiates with the CSP and, if the CSP agrees the negotiation, the CSU continues with the existing service or else searching for new service.

III. Result and Discussion

Implementation the process of putting a theoretical design into the working system. In our experimental analysis Google's free cloud platform and PerfKitBenchmarker monitoring tool are used for the metric monitoring process. Database management is established using MySQL database. Front end development is implemented by using J2ee. For measuring the metric we use three time intervals a day: Morning, peak hour and night. Though there are many metrics available we use only three impartment ones of our experiment: response time, throughput and latency. The collected metrics values are stored in a MySql database. Successively we fix a threshold value for the response time, throughput and latency and it is 200 milliseconds, 300 mbps and hundred milliseconds respectively. Fig 4 shown the dashboard details of QoS monitoring and management system. If the threshold limit exceeds, the mentoring system reckons it as violation. The CSU dashboard consists of two parts, one of them consists of chart, graph and reports. The other part consists of the facilities for the customer to negotiate and feedback.

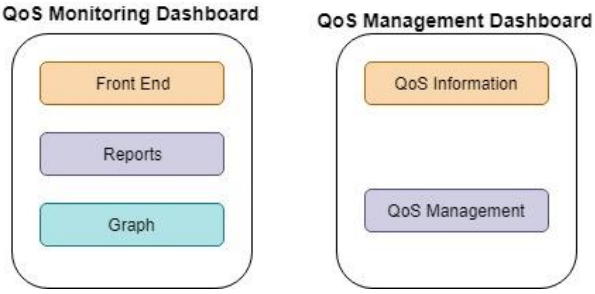

Fig.4 Cloud user dashboard

The preliminary results obtained for detection of SLA violation is shown in fig.5, 6 and 7. The analyzer detects and counts the number of violations and displays them in a graph. The graph contains seven days monitored information. The obtained results and the analysis on these results are as follows: The graphs shows that the violation is much less during the free time and holidays compared with that of the peak hours and the working days. The overall result shows that though CSP provides much resources and computing power, there exists problem in the QOS management. The proposed architecture provides the facilities to the customer to manage the QOS with ease.

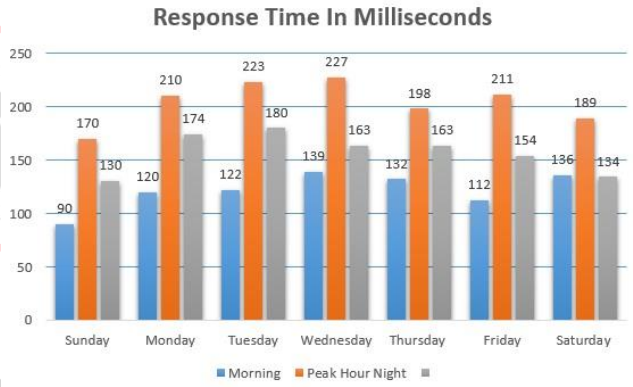

Fig.5 Response time chart

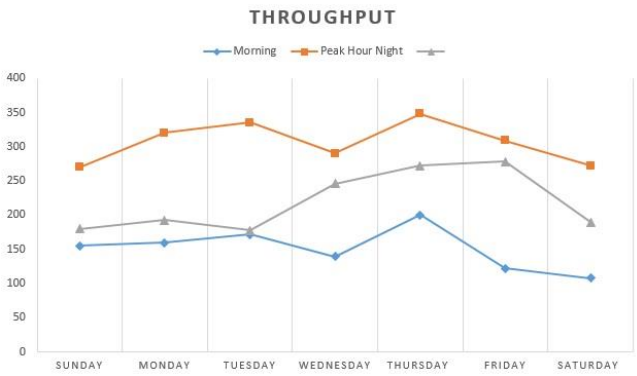

Fig.6 Throughput chart

 DOI : 10.18231/2454-9150.2018.1111

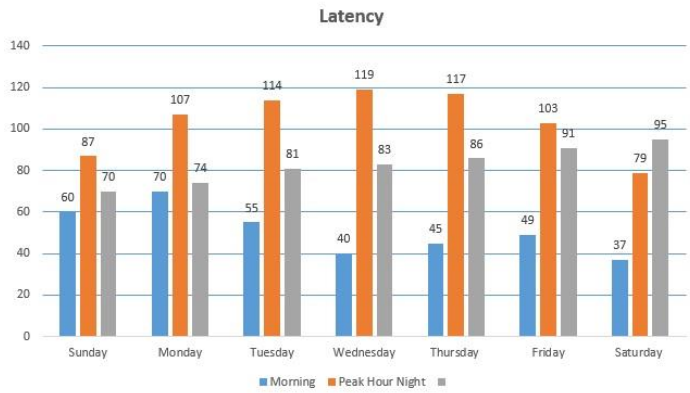

Fig.7 Latency chart

IV. PERFORMANCE ANALYSIS

We compare the performance of the proposed DDA with that of all the already existing three systems such as Self-manageable Case-Based Reasoning Approach, SLA-Based Trust Model Approach and Broker-Based Approach. This comparison is based on the monitoring approach, SLA management and post procedure of SLA violation shown in Table 1.

Table 1. Performance analysis of monitoring SLA system

Source	Monitoring Approach	Real time monitoring	Real time Management	Procedure after SLA violation
[9]	Self-manageable	Yes	No	Not defined
[10]	Self-manageable	Yes	No	Not defined
[11]	Self-manageable	Yes	Yes	Self-handle
[12]	SLA-Based Trust Model	Yes	No	Not defined
[13]	SLA-Based Trust Model	Yes	No	Not defined
[14]	SLA-Based Trust Model	Yes	No	Not defined
[15]	Broker-Based Approach	Yes	No	Not defined
[16]	Broker-Based Approach	Yes	No	Not defined
[17]	Broker-Based Approach	Yes	No	Not defined
	Database Driven Architecture	Yes	Yes	Calculate SLA violation and penalties.

V. CONCLUSION

It is a challenging task to monitor the SLA and prevent SLA violation in cloud environment. An effective SLA management system will improve the confidence level between the cloud service provider and cloud service user. One method of maintaining trust in cloud computing is real-time monitoring of SLA. In this paper, we proposed a Novel Database Driven Architecture for federated cloud services. We divide our proposed DDA architecture into three components: functionality, working attribute and performance analysis. First we proposed an innovative approach to QOS management and monitoring. This approach suits to different types of cloud platforms. Secondly we use a benchmark monitoring tool. It monitors all the metrics in the cloud and store them in the database architecture we proposed. Finally we analyze the performance of the proposed DDA system comparing with the already existing cloud QOS monitoring systems. The experimental result shows that the system we proposed is better than the systems that already exist.

REFERENCES

[1] C. Baudin, B. Martino, and M. Edwards, "Practical Guide to Cloud Service Agreements," Cloud council, Apr-2015. Available: http://www.cloud-council.org/deliverables/cscc-practical-guide-to-cloud-service-agreements.pdf.

[2] Report on Cloud Computing to the OSG Steering Committee", Standard Performance Evaluation Corporation, 2016. Available:

[3] "Cloud Services Industry's 10 Most Critical Metrics." (n.d.).Available: http://guidingmetrics.com/content/cloud-services-industrys-10-mostcritical-metrics. [Accessed 24 June 2017].

[4] "Storage Performance Metrics for the Cloud." Storage, 10 February 2015. Available: http://www.enterprisefeatures.com/storage-performancemetrics-cloud. [Accessed 24 June 2017].

[5] "PerfKitBenchmarker." 11 November 2015. Available: https://en.wikipedia.org/wiki/PerfKitBenchmarker.

[6] L. Wu, S. Garg, and R. Buyya, "Sla-based resource allocation for software as a service provider (saas) in cloud computing environments," in 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2011, pp. 195–204.

[7] Y. O. Yazir, C. Matthews, R. Farahbod, S. Neville, A. Guitouni, S. Ganti, and Y. Coady, "Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis," in 3rd International Conference on Cloud Computing (CLOUD), 2010, pp. 91–98.

[8] D. Ergu, G. Kou, Y. Peng and Y. Shi, "The analytic hierarchy process: task scheduling and resource allocation in cloud computing environment," The Journal of Supercomputing, vol. 64, no. 3, pp. 835–848, 2013.

 DOI : 10.18231/2454-9150.2018.1111

[9] Vincent C. Emeakaroha, Ivona Brandic, Michael Maurer, Schahram Dustdar, "Low level Metrics to High level SLAs-LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments," In: 2010 IEEE International Conference on High Performance Computing & Simulation.

[10] Emeakaroha, V.C, "Towards autonomic detection of sla violations in cloud infrastructures," Future Generation Computer Systems, 28(7), 1017–1029 (2012).

[11] Brandic, Laysi, "A layered approach for sla-violation propagation in selfmanageable cloud infrastructures," In: 2010 IEEE 34th Annual Computer Software and Applications Conference Workshops (COMPSACW). IEEE (2010).

[12] Chandrasekar, A, Mahadevan, M, Varalakshmi, Chandrasekar, K, "QoS monitoring and dynamic trust establishment in the cloud," In: Li, R., Cao, J., Bourgeois, J. (eds.) GPC 2012. LNCS, vol. 7296, pp. 289–301. Springer, Heidelberg (2012).

[13] Daniel D, Lovesum S, "A novel approach for scheduling service request in cloud with trust monitor. In: 2011 International Conference on Signal Processing," Communication, Computing and Networking Technologies (ICSCCN). IEEE (2011).

[14] Alhamad M, Dillon T, Chang E, "Sla-based trust model for cloud computing," In: 2010 13th International Conference on Network-Based Information Systems (NBiS). IEEE (2010).

[15] Badidi E, "A Cloud Service Broker for SLA-based SaaS provisioning," In: 2013 International Conference on Information Society (i-Society). IEEE (2013).

[16] Falasi, A.A, Serhani, M.A., Dssouli, R, "A Model for Multi-levels SLA Monitoring in Federated Cloud Environment," In: 2013 IEEE 10th International Conference on Autonomic and Trusted Computing (UIC/ATC) Ubiquitous Intelligence and Computing. IEEE (2013).

[17] Jrad, F., Tao, J., Streit, A, "SLA based Service Brokering in Intercloud Environments," In: CLOSER (2012).

 DOI : 10.18231/2454-9150.2018.1111