Predictive Modeling of Service Level Agreement Parameters for Cloud Services

SEEMA CHOWHAN, SHAILAJA SHIRWAIKAR
Department of Computer Science, Savitribai Phule Pune University, Pune, India
and
AJAY KUMAR
Department of Computer Science, Jayawant Institute of computer Application, Pune, India

Cloud computing has emerged as an important paradigm in Information and Communication Technology space by enabling cost effective, on demand provisioning of elastic computing resources. With limited or almost negligible upfront investment, lots of organizations are attracted towards cloud, for outsourcing their computational needs. Service Level Agreements (SLA) between Cloud providers and the Cloud users are used to assure Quality of Service (QoS) which is one of the big issues that resists organization from availing cloud resources. SLA management is thus an important activity for Cloud providers as SLA violations may lead to contractual penalties and in turn loss of revenue and customer base. Managing SLA involves constant monitoring and controlling various SLA parameters. Therefore, it is desirable for providers to control possible violations before they happen by predicting the values of SLA parameters using the values continuously measured over a time period. We present an agent based SLA-management with design of a coordinator agent that uses a predictive modeling approach for predicting and mitigating SLA violations. The design is based on a case study on available datasets containing measurements on web services of SLA parameters such as response time and throughput.

Keywords: Service Level Agreement(SLA) , Regression, Supervised learning, Agent based Computing.

1. INTRODUCTION

Cloud is a pool of heterogeneous resources as a mesh of huge infrastructure. Infrastructure refers to both the applications delivered to end users as services over the Internet and the hardware and system software in data centers that is responsible for providing these services. In order to make efficacious use of these resources and ensure their availability to the end users, resource provisioning is done based on certain criteria specified in SLA. Service Level Agreements indicate the service level parameters that are important to ensure Quality of service and specifies agreed levels to these services usually in quantified form. SLA parameter list can vary depending on the customer requirements but usually include higher level attributes such as availability, reliability and low level attributes such as response time, throughput, latency time, downtime per week, Mean time to Repair(MTTR), Mean time between failure(MTBF) etc. [Tang and Tang 2014]. SLA management is extremely important to avoid SLA violations, to ensure Quality of Service and to enforce optimal utilization of cloud resources. As a long term management goal it may help in coming up with improved SLA negotiation plans and Cloud capacity plans. For cloud service providers, short term goal is to prevent SLA violations as much as possible to enhance customer satisfaction and avoid penalty payments. Therefore, it is desirable for providers to continuously measure and monitor SLA parameters and predict possible violations just in time that they can be controlled. However, accurate prediction of quality of cloud services or SLA violation is extremely challenging because QoS of a cloud service fluctuates drastically at small timescales, due to network traffic conditions, cloud platform loads, and other factors. Multi-agent systems (MAS) are a well known approach to model and implement complex distributed systems and applications. Several researchers have proposed Agent based approaches to managing cloud services including SLA management [Sim 2012] [He et al. 2007]. An agent is autonomous soft-

ware that is designed to meet specific objectives by interacting, coordinating and cooperating with other agents [Wooldridge 2009]. This paper presents a simple agent based approach to SLA management at cloud provider level through interaction and co-ordination between three agents: Negotiator, Coordinator and Allocator. The goals and basic functionalities of the coordinator agent along with its interactions with the environment and actions are detailed out. The predictive approach of coordinator agent is designed by using lessons-learned from a case study on available datasets. The paper is organized as follows. Next section presents the background and related work. Section three presents proposed Agent based architecture for SLA-management. Section IV presents predictive modeling case study for throughput and response time as SLA parameters, trained using available datasets. Section V presents the detail design of coordinator agent based on empirical results. The paper ends with conclusion and future directions.

2. BACKGROUND AND RELATED WORK

2.1 Cloud Computing:

Cloud is a parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources, to a set of Customers with heterogeneous requirements, based on Service Level Agreements [Buyya et al. 2001]. SLA is an important document used by customer to judge the quality of service offered and actually provided by service provider. Service provider will greatly benefit from capability of predicting and monitoring SLA violations either by controlling the problem just in time by adding more resources or in revising SLA for future interactions.

2.2 SLA Management in Cloud Environment:

SLA management supervises information about resource capabilities, availabilities and performance during operation, in order to execute actions for services to be delivered according to guarantees agreed with customers. There is then a need for mixing historical, predictive and live information about resources for dynamic re-planning and provisioning of resources. Two typical types of SLA are provider predefined and negotiated SLAs. The provider predefined SLA provides a generic SLA template for all customers. However, customers may have special QoS requirements which may not be included in a predefined SLA. In this case, the customer and the provider will go through negotiation processes to achieve a mutually agreed SLA (Negotiated SLA). When trying to meet clients Service Level Agreement (SLA) for Quality of Service (QoS) and the operating cost, cloud providers are faced with the challenges of under-provisioning and over-provisioning. Under-provisioning often leads to SLA penalty resulting in income loss for cloud provider [Armbrust et al. 2009] [Fang et al. 2012] [Gandhi et al. 2012] and poor Quality of Experience (QoE) for the cloud clients. On the other hand, over-provisioning can lead to excessive energy consumption, high operating cost, and waste of resources.
[Faniyi et al. 2012] presented a distributed simulation approach for cloud federation Middle-ware that is capable of matching cloud users requests with cloud providers offerings without violating cloud users SLA.

2.3 Quality of service parameters (QoS):

Cloud computing focuses on QoS parameters such as response time, throughput, reliability, availability, cost of service etc. QoS parameters play an important role in ranking service providers. QoS parameters are continuously monitored and controlled by service providers to avoid SLA violations. It is reported that VM requires various time durations of boot up time, before it is ready to operate [Imam et al. 2011] [Kupferman et al. 2009] [Lorido-Botrán et al. 2012] [Quiroz et al. 2009]. VM requires 5 to 15 minutes to boot up. It is observed that during this time, system resources are not available, requests cannot be serviced due to insufficient resources, which can lead to SLA violation and penalty on the part of the cloud providers. Provisioning and predicting the need of a VM in advance and making it available just in time can maintain level of avail-

ability and avoid SLA violations. Several top cloud providers (e.g. Amazon, Google, Microsoft) experienced service outages which sometimes lasted for periods ranging from few hours up to one week. The common causes of these SLA violations are unexpected outages caused by software, hardware or network faults [Gunawi et al. 2011]. The uncertainty about the quality of service (QoS) of cloud services reduces user confidence in the technology.

2.4 Agent based Cloud Computing:

The dynamism of cloud, requiring continuous monitoring of requests and resources, handling of ever changing requirements, schedules and prices, selecting appropriate services and plans to meet overall objectives of the cloud, suggests use of autonomous agents for managing cloud. Due to the naturally decentralized architecture, this paradigm provides appropriate concepts for realizing systems that offer inherently non-functional requirements such as scalability, robustness and failure tolerance in cloud. An agent is an autonomous software system that reacts pro-actively to changes in the environment and interacts with other agents, persistently pursuing its goals. The multi-agent system has a set of agents that interact together to resolve a common problem by using the resources and the knowledge base of each agent. For successful interaction agents require ability in terms of human interaction types such as negotiation, coordination, cooperation and teamwork. Cooperation is the process when network of interacting agents exchange their knowledge and capabilities to achieve a common goal. Coordination among agents is about managing their activities. Negotiation is a process by which a group of agents communicate with one another to try to come to a mutually acceptable agreement on some matter.

2.5 Predictive modeling:

Predictive modeling is a collection of techniques that create or extract a model in the form of a mathematical relationship between a set of features from the training data, validate its efficacy by measuring the error or deviation on test data and use it to predict the values of certain features when certain other features are known in the current or future data. With a wide range of modeling techniques available from statistics to machine learning, it has a wide range of applications from spam filtering to Recommender Systems.

2.6 Related work:

SLA violation detection and prediction has been investigated and reported in the literature. Currently, there is limited work in this area for cloud computing but have been extensively used in other related areas such as, Service Oriented Computing (SOC) and web services. For the service provider, important runtime SLA management tasks include (1) monitoring of SLA parameters for resource monitoring and service monitoring to decide if SLAs have been violated in the past (2) analysis of past SLA violations, in which the providers delivery performance is measured against the contract to improve the business process, so that those violations can be prevented for the future (3) prediction of future violations before they have happened.

SLA monitoring is strongly related to QoS monitoring, as SLAs can often be broken down into lower-level QoS metrics for example High level SLA parameter such as service availability into low level parameter such as host up and down time.

With a growing number of alternative Web services there is variation in QoS for the same functionality service. Selecting the best service is a problem for different users. Due to different characteristics like location of service and network environment, the QoS of the same service to different users may be different. Many research efforts have been aimed at predicting the missing QoS values. A user-based collaborative filtering (CF) algorithm is proposed to collect QoS information from different service users, apply similarity mining and then collaborative filtering approach is designed based on the collected QoS data to predict Web service values [Shao et al. 2007]. Zheng present a method based on past usage experiences of service users that follows a collaborative filtering approach for predicting QoS values of Web services and making Web service recommendation [Zheng et al. 2011]. [Chen et al. 2013] proposed a region-based hybrid

CF algorithm to predict the QoS of services. This method discovers the influence of a users location to the accuracy of prediction. Based on hierarchy of regions, these method groups users according to users locations and their QoS records, so that the users in a region are similar. This method does region-wise searching for target user group.
[Liang et al. 2013] proposed a framework for multi-user Web services selection problem. In this prediction method the relationship between QoS attributes is considered. First it predicts the missing multi-QoS values. The method selects the global optimal solution for multiuser by fast match approach according to the historical QoS experience from different users. In this approach, Web service selection problem can be transformed to a maximum weight matching problem by adding virtual Web services according to their processing capacity.
[Sim 2012] has introduced Agent based paradigm for managing cloud services specifically service discovery, negotiation and composition. [He et al. 2007] presented an Agent based framework for solving complex SLA management problems related to SLA formation, recovery and profiling. An multi-agent based negotiation framework is designed by [Chen et al. 2014]used with CloudSim to simulate Resource Allocation. In this paper, we propose an Agent based SLA-management, where the agent uses predictive approach for predicting SLA violations and taking appropriate actions by interacting with other agents. The results on available datasets with throughput and response time measurements on web services are used for understanding the agent percepts and scenarios..

3. MULTI-AGENT BASED ARCHITECTURE FOR SLA-MANAGEMENT

The appropriateness of Agent based approach to handle dynamism in cloud environment is evident and pursued by many researchers [Sim 2012] [He et al. 2007] [Chen et al. 2014]. Here we consider the problem of SLA management at Cloud provider level to avoid SLA violations by predicting the levels of different SLA parameters based on continuous data collection. The process and guidelines provided by Promethus methodology [Padgham and Winikoff 2005] is used in designing and specifying the Agent based Architecture.

3.1 Identifying the system goals

The main goal of the system is optimum utilization of cloud resources avoiding SLA-violations. These goals can be divided into several sub goals as given below

(1) Identifying appropriate SLA template depending on the consumer needs.
(2) Initiating the resource provisioning based on SLA template.
(3) Allocation of resources.
(4) Logging resource utilization parameters.
(5) Predicting SLA-parameters using appropriate prediction model and resource utilization parameters.
(6) Predicting possibility of SLA violation.
(7) Constructing prediction models.
(8) Choosing model depending on SLA parameter and the task.
(9) Modifying or refining SLA.
(10) Requesting changes in Resource allocation.
(11) Adding, Modifying and Deleting SLA templates.

3.2 Identifying different types of agents

The large number of sub goals need to be distributed among different agent types. The sub goals are grouped to give rise to three types of agents: SLA Negotiator Agent, SLA Coordinator agent and Resource Allocator agent. The template for each agent type and the relationships are depicted in Figure 1.

3.2.1 *SLA Negotiator Agent:.* The Agent selects a predefined SLA template from SLA database for its customers. However, customers may have special QoS requirements which may not be included in a predefined SLA template. In this case, the negotiator agent will go through a negotiation process with the customer to achieve a mutually agreed SLA (Negotiated SLA). In order to ensure the agreed SLA, Negotiation agent requires strategies to manage resources to satisfy the QoS specified in SLA. The negotiation agent interacts with Coordinator agent to get information about the resources, at the same time it collects information about SLA template from SLA database and QoS terms, negotiation strategy and service information from knowledge data base.

3.2.2 *SLA Coordinator Agent:.* It receives request from negotiation agent to initiate SLA as also interacts with Resource Allocator agent to collect data about resources and allocate resources. The coordinator agent improves the system efficiency by mapping user QoS parameters to low level system requirements and use predictive information collected from predictive modeling approach to avoid SLA violations by either interacting with Resource Allocator agent for adjusting resources or with Negotiator agent for adjusting SLA. The Coordinator agent can add, modify and delete SLA templates thus restricting the range of available templates depending on the resource availability. Thus Co-ordinator agent plays a very important role of decision making for achieving the system goals of optimal utilization of system resources.

3.2.3 *Resource Allocator Agent:.* Resource allocator agent does resource provisioning as requested by Coordinator agent and also allocation of resources. It collects data about resources from infrastructure layer, continuously monitors resources to find their availability and carries out scheduling of resources.

3.2.4 *SLA Base:.* SLA base stores negotiated service level agreements and SLA templates.

3.2.5 *Knowledge Base:.* The repository stores resource information, prediction parameters, prediction Model, request and resource mapping, resource utilization logs for different SLA parameters. SLA parameters include the providers predefined parameters and the customer specified QoS Parameters.

The three agents and their interactions with each other and the environment basically the cloud infrastructure are depicted in Figure 1.

At detailed design level, for identifying the internals of co-coordinator agent as to how it will accomplish its tasks, we have used an available dataset to enact the predictive modeling approach as discussed in next section.

4. PREDICTION OF SLA PARAMETERS ON AVAILABLE DATASET

The resource Allocator agent that handles resource allocation is also responsible for measurement and continuous data collection of various SLA parameters and makes it available to the co-coordinator agent. For understanding the behavior of co-coordinator agent and the different scenarios, we present a case study of using prediction approach on available dataset. In this paper, the dataset used comes from WS-DREAM (Web Service QoS Datasets). The traces are real-world QoS evaluation results from 64 users on 4532 Web services and includes response-time and throughput values for 65 time instances. A lot of preprocessing is done on this dataset to convert it into appropriate format for analysis. Missing values for some of time instances are replaced with average of adjacent values.

4.1 CHOOSING SLA PARAMETERS

Service Level Agreements indicate different service level parameters that are important to ensure Quality of service. The main task of co-coordinator agent is monitoring and predicting their levels and comparing them with agreed levels in SLA to initiate necessary action to avoid SLA violations. SLA parameter list is usually quite extensive depending on user requirements. In

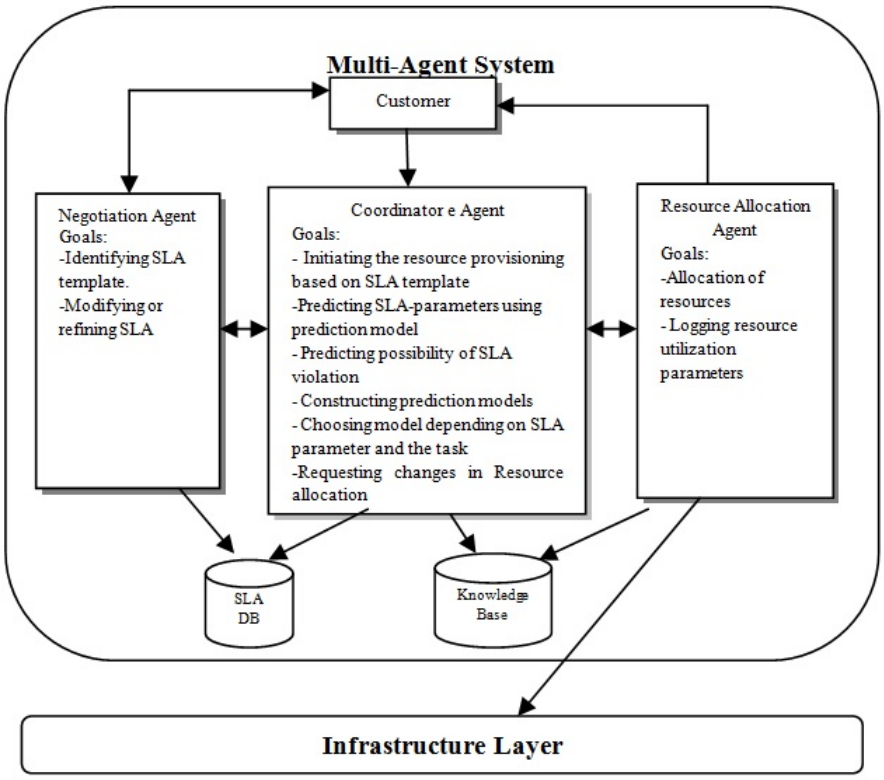

Figure 1 Multi-agent Architecture for cloud

this paper, we have considered throughput and response time. Throughput is the transactions per second and a typical enterprise application will have lots of users each carrying out lots of different transactions. Throughput is used in identifying the regular workload of an application and also as a measure of network performance. Throughput is a critical factor for cloud-based software applications involving video data, scientific data, data being streamed by 'Internet of things' devices, or 'real time' big data systems [1]. All the services mentioned above are throughput oriented. Response time is the total amount of time it takes to respond to a request for service. Response time is not only crucial for small applications but also for larger applications that are running on a public cloud. Cloud data centers have the responsibility to provide the quality of services, despite the dynamic nature of the cloud where the load varies all of a sudden to fulfill the quality requirement, applications hosted on cloud need to be checked for their performance i.e. response time and throughput so that performance factors are within the tolerance limit.

4.2 PREDICTOR AND RESPONSE WINDOW

The total time slots are divided into two windows, the first and the largest window of 41 time slots called training window is used to provide the predictor variables and 4 different size overlapping time slots (Fig 2) of 4, 8, 12 and 24, called testing windows, are used for providing the response variables. The first training window of size 4 is used in model construction while the other windows are used for cross validation. The response variable for each testing window is the mean of throughput values.

[1] https://cloudstore.interoute.com

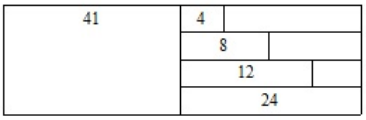

Figure 2 Predictor and Overlapping response windows

4.3 FEATURE SELECTION

The predictor window is used for training and a small set of features need to be selected that essentially describe the training window of 41 throughput values at different time instances. The features should be chosen so that the center of the data set as well as the dispersion of data is captured. There can be several ways of choosing such features and two different approaches are considered for study.

4.3.1 *Five Number Summary.* The five point summary is a descriptive statistics that provides concise information about a large set of observations. The five number summary is extracted from training window which includes Minimum, First quartile, Median, Third quartile and Maximum. It consists of the five most important sample percentiles: The five-number summary gives information about the location (from the median), spread (from the quartiles) and range (from the sample minimum and maximum) of the observations. The above measures are holistic but not algebraic but are computationally expensive as o (n) comparisons are required to compute new median.

4.3.2 *Mean and Variance.* The most commonly used pair of features is mean and variance where mean is a measure of central tendency and the variance captures the dispersion of data.

- Mean is an effective measure of center of a set of data that presents the average throughput of the predictor window. It is an algebraic, distributive measure which can be efficiently computed but it is sensitive to the presence of extreme values.

- The Variance of n observations $x_1, x_2, ..., x_n$ is the average of square of spread of each throughput value about the mean throughput. It is an algebraic measure that can be computed using distributive measures.

Both Mean and variance are scalable and can be easily computed incrementally as the predictor window expands [Finch 2009]. By using following formulas

$$\mu_n = \mu_{n-1} + \frac{x_n - \mu_{n-1}}{n} \tag{1}$$

$$S_n = S_{n-1} + (x_n - \mu_{n-1}) + (x_n - \mu_n) \tag{2}$$

4.4 PREDICTION ALGORITHMS

Several algorithms are available in the literature for predictive modeling with varying degree of accuracy and computational complexity. For comparative study some simple statistics based predictive methods as also Regression analysis are chosen.

4.4.1 *Simple Statistics based Algorithms.* Three simple statistics based algorithm are considered which uniformly aim to predict the mean value of the future interval, based on the values in the predictor window.

- Last-State Based Method (LSAM): The last recorded value in the predictor window will be used as the predicted mean value for the future period.

- Simple Moving Average Method (SMAM): The mean value of the predictor window will be used as the predicted value of the future window.

- Weighted Moving Average Method (WMAM): The weighted mean value of the predictor window will be considered as the predicted mean value for the future. The weight decreases as the time slot grows older. The throughput value of the window is calculated using the following equation.

$$Fwmam = \frac{\sum_{i=1}^{d} ix_i}{\sum_{i=1}^{d} i} \tag{3}$$

4.4.2 *Regression Analysis.* Regression Analysis is a statistical method for extracting mathematical relationship between a set of predictor variables and a response or target variable. In a regression method, a hypothesis is formulated about the relationship between the variables. The available data is used to check the validity of the hypotheses by measuring the error. In Multiple or Multivariate Linear Regression, the hypotheses is a linear relationship between a set of n independent or predictor variables X=$(x_1, ..., x_n)$ and one dependent or response variable y given by

$$y = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 ..... + \Theta_n x_n$$

To estimate the parameters

$$\Theta = \Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 ..... + \Theta_n x_n$$

that gives the best fit, supervised machine learning algorithm minimizes the error function usually computed as the Mean Square Error (MSE) between the predicted and actual values

$$MSE = \frac{1}{2n} \sum (\Theta X^{tranpose} - Y)^2 \tag{4}$$

The gradient descent approach iteratively modifies the parameters by adding the gradient

$$(grad^i)$$

which is the partial derivative of the error with respect to the parameter i.

$$grad^i = \frac{1}{n} \sum (\Theta X^{tranpose} - Y) x_i \tag{5}$$

To avoid over-fitting of the model to the training data, regularization term governed by lambda is added both to the gradient and cost function.

$$MSE = MSE + \frac{lambda}{2n} \sum_{i=1}^{n} (\Theta_i^2) \tag{6}$$

$$grad^i = grad^i + \frac{lambda}{n} \sum_{i=1}^{n} (\Theta_i) \tag{7}$$

The learning curves for training and cross-validation error for different values of lambda is plotted to make the right choice of regularization parameter. For five points summary method the curves for the different training windows are shown in Figure 3(a) and the learning curves for the mean variance method are shown in Figure 3(b)

There are several other algorithms for constructing the prediction model but the efficacy of the algorithm depends on the dataset.

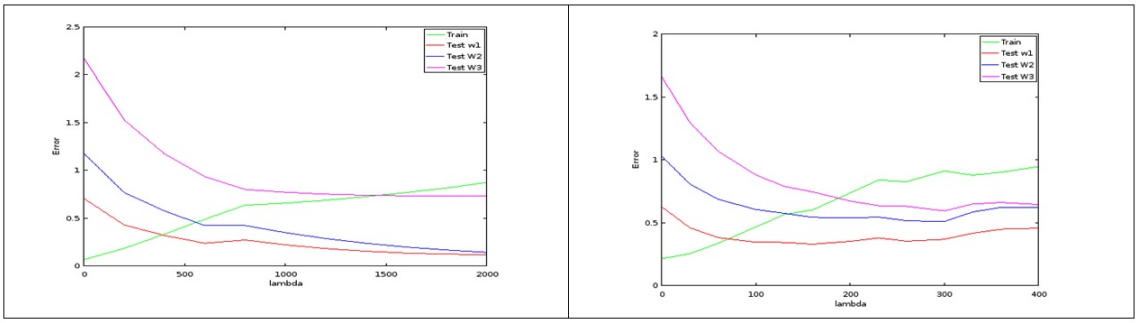

Figure 3: Learning Curves for different regularization Parameters

4.4.3 *Choosing the right Prediction Model.* A comparative analysis of prediction error for different algorithms can be used in deciding the right prediction model. The prediction error for the different algorithms for throughput values on the dataset of 101 web services is given in the following table-I.

Method	CV- W4	CV- W8	CV- W12	CV- W24
LSBM	4.385131	2.720444	2.630262	0.917974
SMAM	0.286616	0.601404	0.973048	1.4322
WMAM	0.678444	0.472896	0.706856	0.720119
RFPS	0.489302	0.23819	0.422686	0.934673
RMV	0.736458	0.352008	0.537237	0.67218

Table I: Prediction error for different algorithms

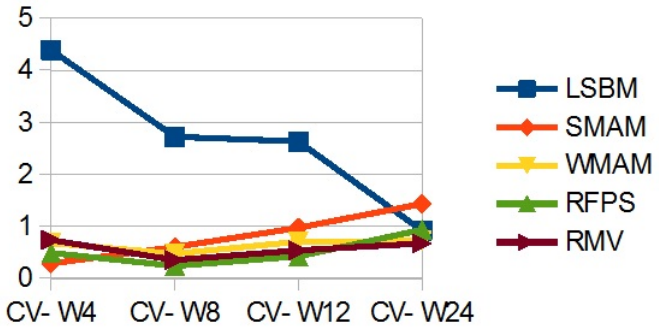

Figure 4 Prediction error Graphical comparisons

The two regression based algorithms give a lesser prediction error in comparison with other prediction algorithms. The two Moving average methods are pretty close compared to computationally inexpensive Last state based method. Between the two regression algorithms the FPS has an upper hand as seen by the error for each web service as shown in the graph below (Figure 5)

For response time the results are completely different as shown in Table-II indicating the supremacy of statistics based (WMAM) method over regression Analysis.

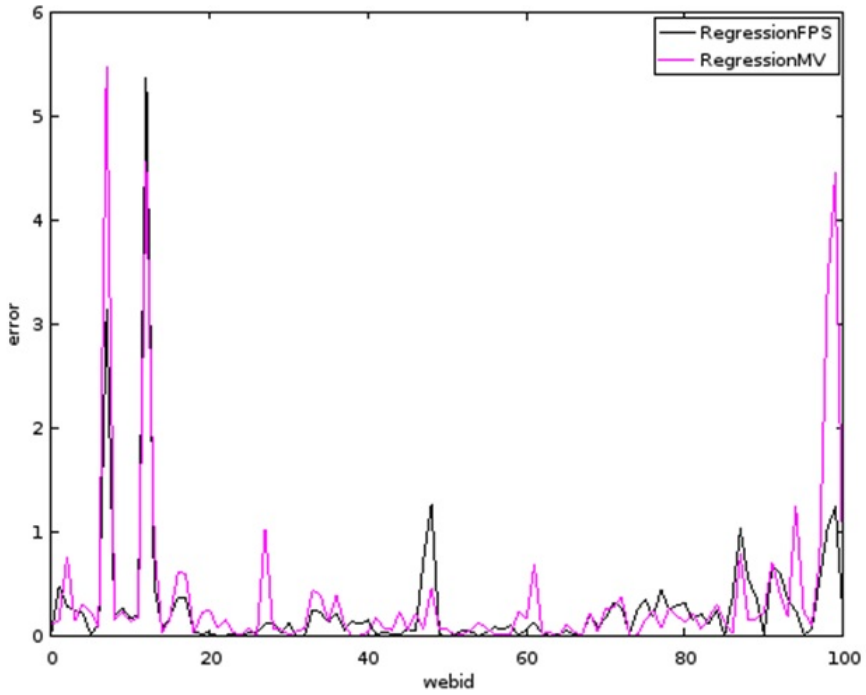

Figure 5 Prediction Error for different web services

Method	CV- W4	CV- W8	CV- W12	CV- W24
LSBM	0.639144	1.852525	2.969395	2.473809
SMAM	0.018163	0.245578	0.769396	0.502252
WMAM	0.018097	0.261045	0.793632	0.526031
RFPS	0.311185	1.196463	2.12404	1.706588
RMV	0.088058	0.443126	1.041621	0.766152

Table II: Prediction error on Response Time

Method	CV- W4	CV- W8	CV- W12	CV- W24
SMAM	8.246241	10.703293	9.744758	12.382277
WMAM	4.978933	5.93329	5.357873	7.052864
RFPS	4.316896	3.522827	2.699616	5.452443
RMV	5.313113	4.204583	3.003469	5.679558

Table III: Prediction error on lager sample size

4.4.4 *Refining the Prediction Model.* Since a single set of parameters are obtained using a training set of web services, the prediction error increases as the number of training samples increase. The following table-III shows the prediction error for throughput data for four prediction methods for a sample size of 300.

However the prediction methods are better placed and between the two prediction methods FPS based algorithm has less prediction error.

Instead of choosing a single set of theta parameters for all the services, the services can be grouped into clusters based on the similarity between their features. For each cluster, a separate set of parameters can be generated which can further improve prediction accuracy. For the sample set of 300 web services k-means clustering was used to form k clusters. The similarity between web services was computed using the two features mean and variance. The k value was chosen after comparing the predictive error. For each cluster a separate set of theta parameters were

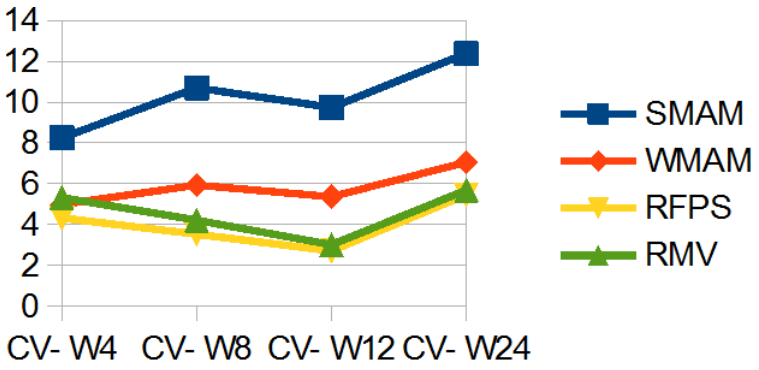

Figure 6 Predictive errors for a larger sample

generated and the prediction error thus gets reduced as shown in the table-IV.

Method	CV- W4	CV- W8	CV- W12	CV- W24
SMAM	8.246241	10.703293	9.744758	12.382277
WMAM	4.978933	5.93329	5.357873	7.052864
RFPS	3.454553	2.985054	2.871744	4.343057
RMV	3.53887	2.960159	2.716925	4.008057

Table IV: Predictive error using clustering of training data

The lessons learned from these experiments are used in designing the internals of Co-coordinator agent.

5. DETAILED SPECIFICATION FOR CO-COORDINATOR AGENT

To understand the functionality of co-ordinator agent, the different interaction scenarios are considered. The interaction diagram depicted in Figure 7 shows the scenario where customer request for new service level agreement is handled at cloud provider level. The request is satisfied by choosing appropriate SLA templates which in a way reflect the current status of available resources. Customer requests for modifying certain SLA parameters can be met by initiating changes in resource status. This will also lead to modifications to existing SLA templates or Coordinator agent can come up with new SLA templates based on availability of resources and add them to SLA DB. The interaction with the consumer (agent) is handled by negotiator agent by continuously interacting with co-ordinator agent and using the knowledge stored in SLA database.

The routine customer service requests are handled by co-ordinator agent by checking their validity in SLA database and passing them over to Resource Allocator agent which actually takes the necessary action for servicing the requests. However periodically the co-ordinator agent uses the predictive modeling to predict the SLA parameters depending on which it may take one of the following actions. i) It may negotiate with Resource allocator agent to modify resource provisioning status to avoid SLA violations. This may lead to modification of SLA templates. ii) It may negotiate with Consumer using the services of Negotiator agent, to initiate modifications to SLA levels to avoid SLA violations. This may lead to modifications to SLA templates. iii) It may use the predicted values to add, modify or delete SLA templates The co-ordinator agent may also periodically construct new prediction models based on the current data logged by the Resource allocator agent. Some of these scenarios are depicted in Figure 3. The power of Co-ordinator agent can be further enhanced by adding new prediction strategies.

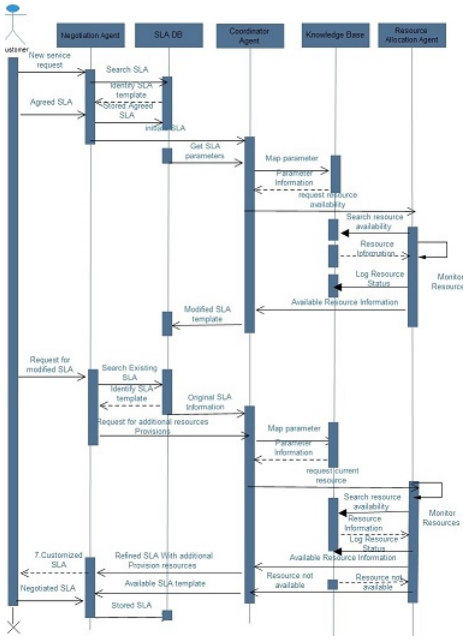

Figure 7 New Service Request

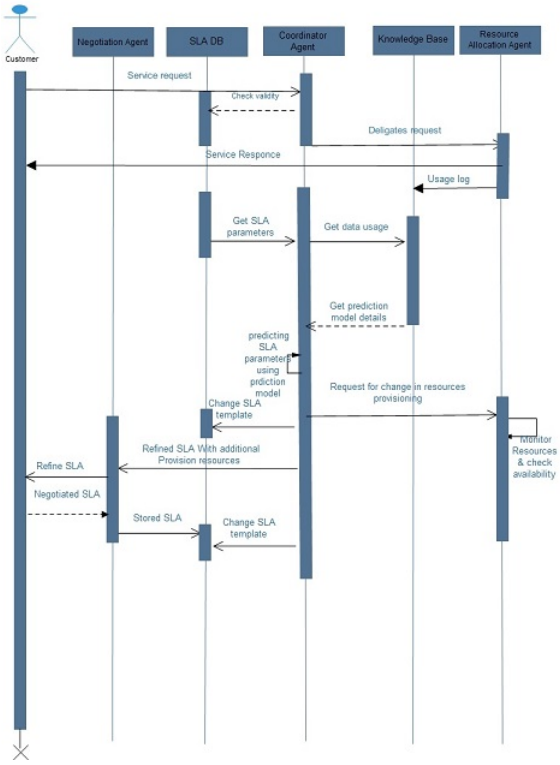

Figure 8 Routine Customer Service Request

6. CONCLUSION AND FUTURE WORK

Cloud data centers are widely utilized for the provisioning of resources. To provide better performance, reliability and availability before provisioning of new resources it is desired that existing resources are used to their optimum level, at the same time provider also has to use prediction model for predicting SLA violation before they occur for better performance. The SLA violations can be averted by continuous monitoring and controlling various parameters specified in the SLA. Predictive modeling approach improved by clustering as demonstrated in this paper can help in managing SLA violations as well as preparing SLA negotiation plans. This paper presents agent paradigm for designing and constructing multi-agent model for effective cloud resource management. The proposed multi-agent based SLA negotiation model applied to cloud resources at cloud provider level fits well into the more general schemes described in [19] at cloud level. The predictive analytics can be further enhanced to increase the decisive power of co-ordinator agent. The predictive strategy depends on the chosen SLA parameter and there are several other important SLA parameters that need to be studied and wide range of effective predictive algorithms. In the future, we plan to implement the model by choosing the appropriate agent based development platform. The simulation experiments carried out with available data can be used to further enhance the model.

REFERENCES

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., and Zaharia, M. 2009. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28 (Feb), EECS Department, University of California, Berkeley.

Buyya, R., Abramson, D., and Giddy, J. 2001. A case for economy grid architecture for service-oriented grid computing. In *IPDPS*, Volume 1, pp. 20083–1.

Chen, J., Han, X., and Jiang, G. 2014. A negotiation model based on multiagent system under cloud computing. In *The Ninth International Multi-Conference on Computing in the Global Information Technology*, pp. 157–164.

Chen, X., Zheng, Z., Liu, X., Huang, Z., and Sun, H. 2013. Personalized qos-aware web service recommendation and visualization. *IEEE Transactions on Services Computing 6,* 1, 35–47.

Fang, W., Lu, Z., Wu, J., and Cao, Z. 2012. Rpps: a novel resource prediction and provisioning scheme in cloud data center. In *Services Computing (SCC), 2012 IEEE Ninth International Conference on*, pp. 609–616. IEEE.

Faniyi, F., Bahsoon, R., and Theodoropoulos, G. 2012. A dynamic data-driven simulation approach for preventing service level agreement violations in cloud federation. *Procedia Computer Science 9*, 1167–1176.

Finch, T. 2009. Incremental calculation of weighted mean and variance. *University of Cambridge 4*, 11–5.

Gandhi, A., Chen, Y., Gmach, D., Arlitt, M., and Marwah, M. 2012. Hybrid resource provisioning for minimizing data center sla violations and power consumption. *Sustainable Computing: Informatics and Systems 2,* 2, 91–104.

Gunawi, H. S., Do, T., Hellerstein, J. M., Stoica, I., Borthakur, D., and Robbins, J. 2011. Failure as a service (faas): A cloud service for large-scale, online failure drills. *University of California, Berkeley, Berkeley 3.*

He, Q., Yan, J., Kowalczyk, R., Jin, H., and Yang, Y. 2007. An agent-based framework for service level agreement management. In *2007 11th International Conference on Computer Supported Cooperative Work in Design*, pp. 412–417. IEEE.

Imam, M. T., Miskhat, S. F., Rahman, R. M., and Amin, M. A. 2011. Neural network and regression based processor load prediction for efficient scaling of grid and cloud resources. In *Computer and Information Technology (ICCIT), 2011 14th International Conference on*, pp. 333–338. IEEE.

Kupferman, J., Silverman, J., Jara, P., and Browne, J. 2009. Scaling into the cloud. *CS270-advanced operating systems*.

Liang, Z., Zou, H., Guo, J., Yang, F., and Lin, R. 2013. Selecting web service for multi-user based on multi-qos prediction. In *Services Computing (SCC), 2013 IEEE International Conference on*, pp. 551–558. IEEE.

Lorido-Botrán, T., Miguel-Alonso, J., and Lozano, J. A. 2012. Auto-scaling techniques for elastic applications in cloud environments. *Department of Computer Architecture and Technology, University of Basque Country, Tech. Rep. EHU-KAT-IK-09-12*.

Padgham, L. and Winikoff, M. 2005. *Developing intelligent agent systems: A practical guide*, Volume 13. John Wiley & Sons.

Quiroz, A., Kim, H., Parashar, M., Gnanasambandam, N., and Sharma, N. 2009. Towards autonomic workload provisioning for enterprise grids and clouds. In *2009 10th IEEE/ACM International Conference on Grid Computing*, pp. 50–57. IEEE.

Shao, L., Zhang, J., Wei, Y., Zhao, J., Xie, B., and Mei, H. 2007. Personalized qos prediction forweb services via collaborative filtering. In *IEEE International Conference on Web Services (ICWS 2007)*, pp. 439–446. IEEE.

Sim, M. K. 2012. Agent-based cloud computing. *IEEE Transactions on Services Computing Vol.5,* 4.

Tang, B. and Tang, M. 2014. Bayesian model-based prediction of service level agreement violations for cloud services. In *TASE*, pp. 170–176.

Wooldridge, M. 2009. *An introduction to multiagent systems.* John Wiley & Sons.

Zheng, Z., Ma, H., Lyu, M., and King, I. 2011. Qos-aware web service recommendation by collaborative filtering. *IEEE Transactions on Service Computing Vol.4*, pp.140152.

Ms. Seema Chowhan is working as a faculty and head in subject of computer science in Baburaoji Gholap College Pune, India affiliated to Savitribai Phule Pune University, Pune.She has 17+ years of experience in teaching UG and PG courses.She has completed M.Phil(CS) Her research interests include Cloud Computing and Networking.

Dr. Shailaja Shirwaikar has a Ph. D. in Mathematics of Mumbai University, India and worked as Associate Professor at Department of Computer Science, Nowrosjee Wadia College affiliated to Savitribai Phule Pune University, Pune for last 27 years. Her research interests include Soft Computing, Big Data Analytics, Software Engineering and Cloud Computing.

Dr.Ajay Kumar experience covers more than 26 years of teaching and 6 years of Industrial experience as IT Technical Director and Senior Software project manager. He has an outstanding academic career completed B.Sc. App. Sc. (Electrical) in 1988, M.Sc. App. Sc. (Computer Science-Engineering and Technology) in1992 and PhD in1995. Presently, working as Director at JSPMs Jayawant Technical Campus, Pune (Affiliated to Pune University). His research areas are Computer Networks, Wireless and Mobile Computing, Cloud computing, Information and Network Security. There are 74 publications at National and International Journals and Conferences and also worked as expert, appointed by C-DAC to find Patent-ability of Patent Applications in ICT area. Six commercial projects are completed by him for various companies/ Institutions. He holds variety of imperative position like Examiner, Member of Board of Studies for Computer and IT, Expert at UGC.