http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (20): 5885-5890, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.5885.5890

Research on Performance Enhancement of Cloud Based on Scheduling and Management Mode

[1]Bin Chen, [1]Zhijian Wang, [2]Mingrong Mao and [2]Ningze Shen
[1]Computer and Information Engineering College of HoHai University, Jiangsu, Nanjing, 211100, China
[2]Information Office of Nanjing Normal University, Jiangsu, Nanjing, 210046, China

Abstract: Job dispatching and resource management are the most import factors which decide the performance of the cloud computing environment. Job dispatching can be devide into User dispatching and Task dispatching based on dispatching model. There are vatious significant issues in resource management, such as maximum computing performance and green computing, aim to accomplish tasks with the lowest cost. Dispatching and management objects must be collected by system statistic information as reference. The Performance Agent and Server Interface Method (PASI) has been collected the row data for the cloud. The commom security technology applied in the cloud are passive defense methods, real-time active monitoring or Deep Packets Inspection (DPI). The main purpose is to realize active inspection and defense. This study proposes an effective framework to enhance comprehensive performance guideline of cloud computing center. The framework is built based on the PASI integrate with DPI module and put forward a job dispatching and resource management module based on safe PASI. Results of the experiment indicate that the model effectively improve the efficiency of the cloud environment.

Key words: Cloud computing, queuing, performance analysis, collection model

INTRODUCTION

Cloud computing architecture makes building high-quality application by using large pools of distributed components to be possible. It is obviously to become an indispensably part of the enterprises which engaged to optimize their budget on services for obtain the services of application, information and storage and computing capacity. As the increase on requirement and complexity of modern cloud computing framework, performance analysis and management issue cause significantly attention on the trade union.

As there is little consensus on how to define the cloud computing, there are still a few common key points in the definition. This new cloud computing software model is a shift from the traditional single tenant approach to software development to that of a scalable, multi-tenant, multi-platform, multi-network and global (Foster *et al*., 2008). From their opinion Above all, cloud computing is a specialized distributed computing paradigm. It differs from traditional ones in that (1) it is massively scalable, (2) can be encapsulated as an abstract entity that delivers different levels of services to customers outside the Cloud, (3) it is driven by economies of scale and (4) the services can be dynamically configured (via virtualization or other approaches) and delivered on demand (Shao *et al*., 2007). As Cloud defined on services, it should be provided stable and smoothly service and avoid prompt execution caused by request message task exceed the response capacity of the cloud center.

As the descriptions mentioned above, providing enough performance information of cloud component is essential and build high-quality cloud services obviously becomes an important and necessary research category. Mutual information is exploited to quantify the relevance and redundancy among the large number of performance metrics (Zhang *et al*., 2012). However, continuous monitoring and large system scale lead to the overwhelming volume of data collected by health monitoring tools. So collect exactly on sufficient and valid data for performance analysis is necessary. It means that, appropriate selection method and collection frequency of working data is very important for the effect and exactitude of the performance analysis.

As a cloud application typically consists of multiple cloud components communicating with each other over application programming interfaces (Fu, 2011), In this study, we propose a system performance collection framework construct on client and server mode. We

Corresponding Author: Bin Chen, Computer and Information Engineering College of HoHai University, Jiangsu, Nanjing, 211100, China

obtain reference Parameter from Performance Agent (PMA) deployed on the different cloud component Performance Client (PMC) and recycle the data to the Performance Server (PMS) deployed on the cloud task queue management cluster.

The rest of this study is organized as follows: Section II describes the related work about performance analysis of the cloud computing center. Section III presents the PASI model for cloud computing on IAAS in detail. Section IV provides the modeling and analysis. Section V discusses the numerical illustration and Section VI concludes the study while give the perspective about the future work.

RELATED WORK

Performance analysis has become one of the hint point research field in cloud computing recent years. The research works on the performance analysis of cloud can be almost divided into three categories by research object: Performance analysis on cloud center, Performance analysis on cloud application and Performance analysis on cloud components. And can be recognized as two aspects: Prediction and real time status (RajaRaajeswari *et al*., 2012).

Performance of cloud center was described in a new approximate analytical model and analyzed under burst arrival and total rejection policy (Khazaei *et al*., 2011). In the study model the cloud environment as an M[x]/G/m/m+r queuing system. As a result The number of facility nodes is m and the capacity of system is equal to m+r.

It was proposed a typical cloud environment comprises thousands of clouds servers (physical machines) and each of them can be segmented into a number of independent or interconnected VMs (Virtual Machines) (Zhang *et al*., 2011) and services can be distributed across different VMs as well as physical machines within a cloud as well as distributed clouds. And the study focuses on the measurement of the performance analysis of cloud application (Jain, 2005). Porting a scientific application like the SN factory pipeline to the Amazon EC2 framework requires the development of some infrastructure and significant planning and testing (Jackson *et al*., 2011).

As analysis aspect discussed above is more macroscopically, expound a special performance analysis method towards cloud applications and components. As computational grids process large, computationally intensive problems on small data sets (Bell *et al*., 2002), it shows the implementation of the complicated functions by combing several abstract tasks. Each task can selects an optimal cloud component from a set of functionally equivalent component candidates. The cloud application is implemented for task execution by composing the selected components which candidates are distributed in different locations and invoked through communication links. The study proposed a collector and predictor model for performance monitor which monitored the user-side real-time performance of cloud components. All of them aim to gain more accurate analysis result of the performance of the cloud.

In this study, we focus on the collection of related performance data on PASI model. Compare the integrality and validity of the performance data collected by the model and before, we prove the usability of the model.

PASI MODEL FOR CLOUD COMPUTING

The architecture of performance client, agent and server collection model established on the deployed of application on the portal, host and VMs of the cloud. The PMS application embedded on the portal of the cloud, it is consist by the map of host queue which include the performance information of the hosts and the VMs run on them. Consumers need to make prediction on quality of unused web services before selecting. Usually, this prediction is based on other consumers' experiences (Ke *et al*., 2010).

As Figure 1, the performance data is collect from the agent application deployed on the hosts and the data of the agent comes from the client application deployed on the VMs. Because all the available cloud components from user-side for evaluation purpose is expensive and impractical (Goswami *et al*., 2012), the performance management table on PMS including host index table, host data table, host occupy/idle table and the crucial information placed in the host index table contain VM index table, VM data table, VM occupy/idle table, VM current data table, VM history quarter data table and VM history day data table. The key data lingering on the performance table include: (1) Request times in the period. (2) Vacant times in the period. (3) Average request time in the period. (4) Average failure rate for quest in the period. And the granularity of the data consists by current quarter, current day, history quarter queue and history day queue. Performance data report from the PMC running on the VM and received by PMA deployed in the host. The PMA compute the data during the setup collect period and synchronize to the PMS when the time arrives. The performance data transmitted to the PMS will be put together and generate ultimate statistic information by queuing theory modeling as a result. All PMC embedded in the different VM which distributed on either a single

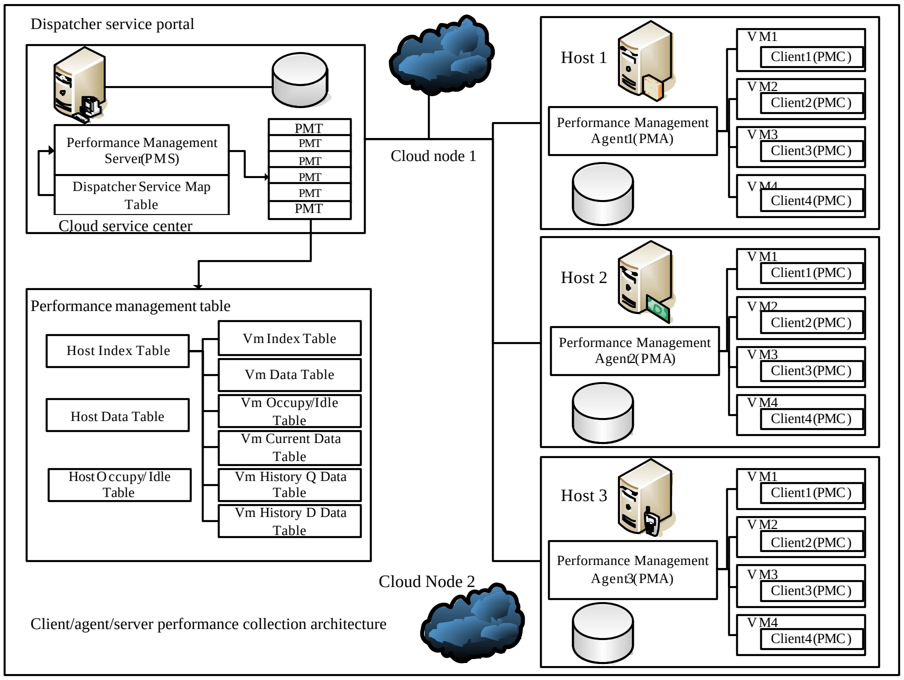

Fig. 1: Architecture of performance client, agent and server collection model

cloud node or multiple cloud nodes that the PMA run on. Successful provision of infrastructure-as-a-service (IaaS) and, consequently, widespread adoption of cloud computing necessitates accurate performance evaluation that allows service providers to dimension their resources in order to fulfill the service level agreements with their customers.

MODELING AND ANALYSIS

We consider the performance data collect from VMs, the cloud component response time is meaning the time from arrival the VM to the ending of response, the time recode by the PMC and relevant information such as average CPU and memory occupancy rate. The PMA receive the data transmit from PMC, then process for statistics. When a quarter arrive, the PMA will packet the total data of the period and transfer to the PMS. The information includes the each fragment of PMC response and handle by the PMA which includes such as: Request times in the period, vacant times in the period, average request time in the period, average failure rate for quest in the period of each corresponding PMC. After the PMS obtain the data from PMA, the data will be accumulated to current day record and synchronized to the history quarter and history day queue if the condition reaches. Of course the common response of user request will be handled real-time. The steady-state probability distributions and the expected number of customers in the system are derived which are used to construct a cost function. It can be described by Fig. 2.

We adopt queuing theory as the abstract model for get the result of average running time of the VM in the period. In order to gain maximum net profit, the threshold values of queue length at which servers are made available one by one, are determined. It is very important to place requested applications into available computing servers regarding to energy consumption. And we use an approximate embedded Markovian process for analyzing the result of the steady VM queue size and the average number of client request in the system with designated transactor. We assume the request arrive VM is abide by Poisson distribution with λ and is subordinated to the exponent of μ. If the request number is i, the service complete rate μ_i:

$$\mu_i = i\mu(1 \le i \le M);\ \mu M(M \le i \le N) \tag{1}$$

According to queuing theory, VM service intensity is $\delta = \lambda/\mu$, the probability for coexists of p_i:

$$p_i = \frac{\delta^i}{i!}p_0(1 \le i \le M); \frac{\delta^i}{M!M^{i-M}}p_0(M \le i \le N) \tag{2}$$

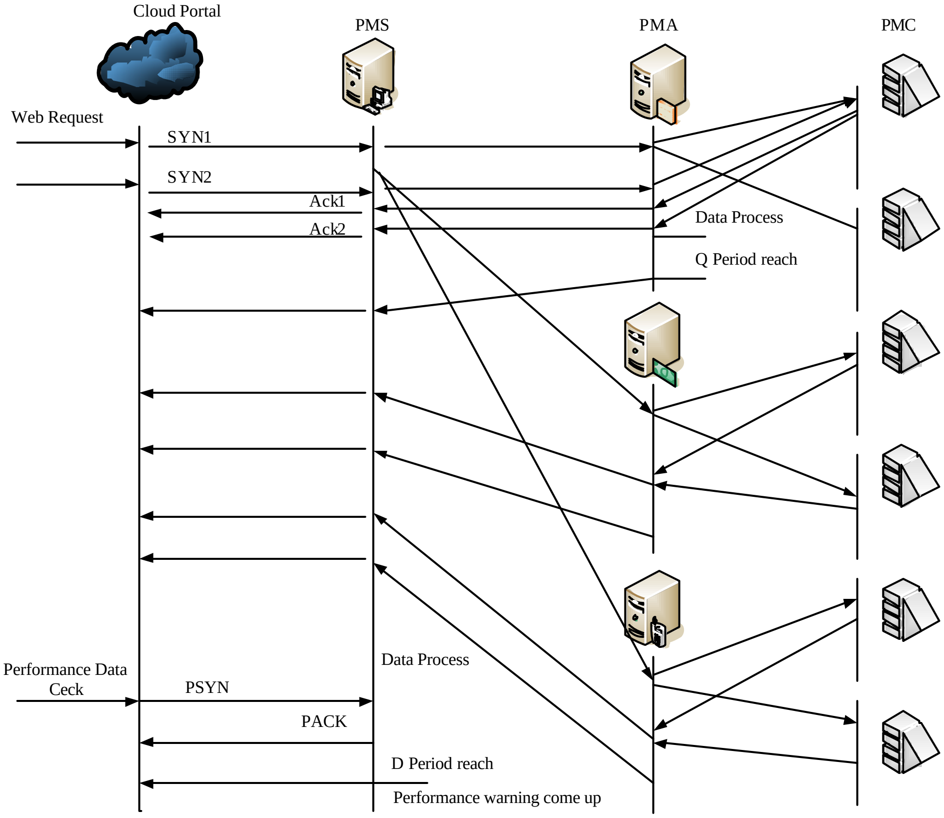

Fig. 2: Web request distribution and performance information organization in CAS

$$p_i = (\sum_{i=0}^{M-1} \frac{\delta^i}{i!} + \sum_{i=M}^{N} \frac{\delta^i}{M!M^{i-M}})^{-1} (i = 0);$$
$$\frac{\delta^i}{i!} p_0 (1 \le i \le M); \frac{\delta^i}{M!M^{i-M}} p_0 (M \le i \le N) \tag{3}$$

$$T_m = \sum_{i=0}^{N} p_i \frac{1}{\mu_i} \tag{4}$$

The result of T_m is the degree of load during the specified period. Then we will research on the length of the request queue transmit to PMC and ultimately feedback to PMS. We assume the requests are random and exponential distribution with mean arrival rate σ and mean service rate θ_i for $i_{th}(1 = i = y)$ PMC. M_i means the specific level of PMC available. We define S_x as the stable probability of the x external requests on the y PMC processing system. The equations for the CAS with y PMC are as:

$$\sigma S_0 = \theta_1 S_1$$
$$(\sigma + \theta_1) S_x = \sigma S_{x-1} + \theta_1 S_{x+1}$$
$$1 \le x \le M_1 - 2 \tag{5}$$

$$(\sigma + \phi_i) S_{M_{i-1}} = \sigma S_{M_{i-2}} + \phi_{i+1} S_{M_i}$$
$$1 \le i \le y - 1, \phi_i = \sum_{j=1}^{i} \theta_j \tag{6}$$

$$(\sigma + \phi_i) S_x = \sigma S_{x-1} + \phi_i S_{x+1}$$
$$2 \le i \le y - 1, M_{i-1} \le x \le M_i - 2 \tag{7}$$

$$(\sigma + \phi_i) S_x = \sigma S_{x-1} + \phi_y S_{x+1}$$
$$M_{y-1} \le x \le K - 1 \tag{8}$$

$$\phi_y S_K = \sigma S_{K-1} \tag{9}$$

From the relationship between stable probability Sx multiple with service rate θ_i and arrive rate σ, we can get the stable length of the requests queue by (7)-(9) as:

$$S_x = \eta_1^0 S_0, 1 \le x \le M_1 - 1$$
$$\eta_j = \sigma / \phi_j \tag{10}$$

$$S_x \left\{ \prod_{j=1}^{i-1} \eta_j^{M_j - M_{j-1}} \right\} \eta_i^{x - M_{i-1} + 1} S_0$$
$$i = 2, 3, \ldots, y - 1, M_{i-1} \le x \le M_i - 1 \tag{11}$$

$$S_x = \left\{\prod_{j=1}^{y-1} \eta_j^{M_j - M_{j-1}}\right\} \eta_y^{x - M_{y-1}+1} S_0 \qquad (12)$$

$$M_{y-1} \le x \le K$$

$$S_0 = \sum_{j=0}^{M_1 - 1} S_j + \sum_{j=1}^{y-1} \sum_{i=M_{j-1}}^{M_j - 1} S_j + \sum_{j=M_{y-1}}^{K} S_j$$

$$= \left[\frac{1-\eta_1^{M_1}}{1-\eta_1} + \sum_{j=2}^{y-1}\prod_{i=1}^{j-1} \eta_i^{M_i - M_{i-1}} \eta_j \left(\frac{1-\eta_j^{M_j - M_{j-1}}}{1-\eta_j}\right) \qquad (13)\right.$$

$$\left. + \prod_{i=1}^{y-1} \eta_i^{M_i - M_{i-1}} \eta_y \left(\frac{1-\eta_y^{K - M_{y-1}+1}}{1-\eta_y}\right)\right]^{-1}$$

The $i_{th}(1 = i = y)$ PMC running in the system can be description by:

$$S(r) = \text{Prob}\{M_{i-1} \le x \le M_i - 1\}$$

$$= \prod\nolimits_{j=1}^{i-1} \eta^{M_j - M_{j-1}} \eta_i \left(\frac{1-\eta_i^{M_i - M_{i-1}}}{1-\eta_i}\right) S_0 \qquad (14)$$

So we can derive that y PMC are running in the cloud:

$$S(y) = \text{Prob}\{M_{y-1} \le x \le K\}$$

$$= \prod\nolimits_{j=1}^{y-1} \eta^{M_j - M_{j-1}} \eta_y \left(\frac{1-\eta_y^{K - M_{y-1}+1}}{1-\eta_y}\right) S_0 \qquad (15)$$

That $i_{th}(1 = i = y)$ PMC keep in working state probability is:

$$S_E(i) = \sum_{j=i}^{y} S(j) \qquad (16)$$

We can obtain the average load degree of the cloud center as (4)+(12):

$$D_{\text{cloud}_{Q/D}} = \frac{T_m * S_E(i)}{P_{Q/D}} \qquad (17)$$

Finally, cloud center portal system can adjust VM resource automatically by relevant method and produce the overload warning to controller system when the D_{cloud} is beyond the threshold.

NUMERICAL ILLUSTRATION

In this section, we will demonstrate the mathematical model loading time in HPL which is called high performance Link pack standard and reckoning of the probability of the x external requests on the y PMC processing system by different configurations. In config1 the PMC receive the request in the rate as σ_j=1 and in config2 as σ_j=1+ (j+1)*P_r, P_r is the deployment coefficient.

We give the T_m as the degree of load during the specified period, numerical results of $S_E(i)$ and $D_{\text{cloud}_{Q/D}}$ for CAS system by Table 1-4. We deploy the different parameter as the condition of the experiment.

In summary, as Fig. 3, we can argue the case of relationship between T_M, S_E and $D_{Q/D}$ effectively.

From the Table 1-4, we can get the conclusion that the change of y, K, σ, σ obviously impact the

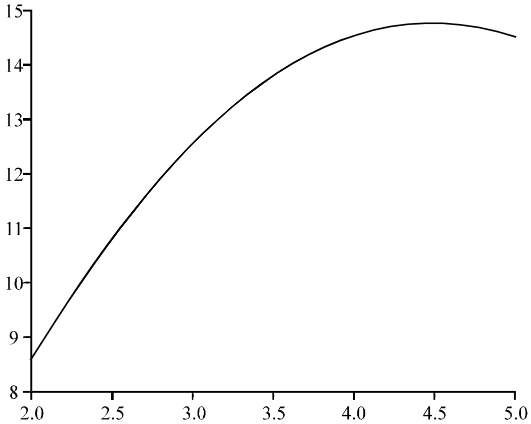

Fig. 3: Compound cloud performance chart

Table 1: Cloud center working load degree (y = 4, K = 20, σ = 4, θ = 65.79)

	Deploy parameter (y = 4, K = 20, σ = 4, θ = 65.79)			
Evl	cfg1 (y)	cfg2 (y)	cfg1 (K)	cfg2 (K)
T_{mQ}	13.32	11.54	12.98	11.21
T_{mD}	1209	1005	1134	982
S_E	0.9302	0.8735	0.9103	0.8501
D_Q	0.8260	0.6720	0.7877	0.6353
D_D	0.7810	0.6096	0.7169	0.5797

Table 2: Cloud center working load degree (y = 4, K = 20, σ = 5, θ = 63.15)

	Deploy parameter (y = 4, K = 20, σ = 5, θ = 63.15)			
Evl	cfg1 (y)	cfg2 (y)	cfg1 (K)	cfg2 (K)
T_{mQ}	14.63	11.97	13.51	12.01
T_{mD}	1308	1097	1217	1032
S_E	0.9415	0.8801	0.9216	0.8673
D_Q	0.9183	0.7023	0.8301	0.6944
D_D	0.8552	0.6705	0.7789	0.6216

Table 3: Cloud center working load degree (y = 5, K = 30, σ = 4, θ = 66.12)

	Deploy parameter (y = 5, K = 30, σ = 4, θ = 66.12)			
Evl	cfg1 (y)	cfg2 (y)	cfg1 (K)	cfg2 (K)
T_{mQ}	12.17	10.01	11.54	9.93
T_{mD}	1132	961	1028	913
S_E	0.9023	0.8311	0.8958	0.8383
D_Q	0.7321	0.5546	0.6892	0.5550
D_D	0.7093	0.5546	0.6395	0.5315

Table 4: Cloud center working load degree ($y = 6$, $K = 30$, $\sigma = 5$, $\theta = 64.07$)

	Deploy parameter ($y = 6$, $K = 30$, $\sigma = 5$, $\theta = 64.07$)			
Evl	cfg1 (y)	cfg2 (y)	cfg1 (K)	cfg2 (K)
T_{mQ}	11.65	11.92	12.70	11.03
T_{mD}	1056	1063	1089	977
S_E	0.9396	0.8743	0.9088	0.8485
D_Q	0.7250	0.6948	0.7695	0.6239
D_D	0.6904	0.6454	0.6877	0.5757

result of $T_{mQ/D}$ and the result of $T_{mQ/D}$ will cause the change of the $D_{Q/D}$.

CONCLUSIONS

In this study, we have described a performance collection model for cloud computing with client and server interface on IAAS mode. In this model, we can gather performance information from client to agent than to server and obtain the average load degree of the cloud center ultimately by them. The cloud center system can adjust the cloud component composition by the portal.

For future work, we will concentrate on back pressure mechanism based on cloud compute performance analysis model and the corresponding adjusts strategy.

ACKNOWLEDGMENTS

The key technologies research of information metadata applications of the project based on volunteer computing supported by NSFC [Supported by NSFC NO. (61103017)].

REFERENCES

Bell, W.H., D.G. Cameron, L. Capozza, A.P. Millar, K. Stockinger and F. Zini, 2002. Simulation of dynamic grid replication strategies in OptorSim. Proceedings of the 3rd International Workshop on Grid Computing, November 18, 2002, Baltimore, MD., USA., pp: 46-57.

Foster, I., Y. Zhao, I. Raicu and S. Lu, 2008. Cloud computing and grid computing 360-degree compared. Proceedings of the Grid Computing Environments Workshop, November 12-16, 2008, Austin, Texas, USA., pp: 1-10.

Fu, S., 2011. Performance metric selection for autonomic anomaly detection on cloud computing systems. Proceedings of the IEEE Global Telecommunications Conference, December 5-9, 2011, Houston, TX., USA., pp: 1-5.

Goswami, V., S.S. Patra and G.B. Mund, 2012. Performance analysis of cloud with queue-dependent virtual machines. Proceedings of the 1st International Conference on Recent Advances in Information Technology, March 15-17, 2012, Dhanbad, India, pp: 357-362.

Jackson, K.R., K. Muriki, L. Ramakrishnan, K.J. Runge and R.C. Thomas, 2011. Performance and cost analysis of the supernova factory on the Amazon AWS cloud. Sci. Program., 19: 107-119.

Jain, M., 2005. Finite capacity M/M/r queueing system with queue-dependent servers. Comput. Math. Appl., 50: 187-199.

Ke, J.B., J.C. Ke and C.H. Lin, 2010. Cost optimization of an M/M/r queueing system with queue-dependent servers: Genetic algorithm. Proceedings of the 5th International Conference on Queueing Theory and Network Applications, July 24-26, 2010, Beijing, China, pp: 82-86.

Khazaei, H., J. Misi and V.B. Misic, 2011. Performance analysis of cloud centers under burst arrivals and total rejection policy. Proceedings of the IEEE Global Telecommunications Conference, December 5-9, 2011, Houston, TX., USA., pp: 1-6.

RajaRaajeswari, S., R. Selvarani and P. Raj, 2012. A performance analysis method for Service-Oriented Cloud Applications (SOCAs). Proceedings of the International Conference on Computer Communication and Informatics, January 10-12, 2012, Coimbatore, India, pp: 1-7.

Shao, L., J. Zhang, Y. Wei, J. Zhao, B. Xie and H. Mei, 2007. Personalized QoS prediction for web services via collaborative filtering. Proceedings of the IEEE International Conference on Web Services, July 9-13, 2007, Salt Lake City, Utah, USA., pp: 439-446.

Zhang, Y., Z. Zheng and M.R. Lyu, 2011. Exploring latent features for memory-based QoS prediction in cloud computing. Proceedings of the 30th IEEE Symposium on Reliable Distributed Systems, October 4-7, 2011, Madrid, Spain, pp: 1-10.

Zhang, Y.L., Z.B. Zheng and M.R. Lyu, 2012. Real-Time performance prediction for cloud components. Proceedings of the 15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, April 11, 2012, Shenzhen, China, pp: 106-111.