Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A SOFTWARE SERVICE MODEL USING SCHEDULE BASED FAIR QUEUE WEIGHT FOR DYNAMIC ADMISSION CONTROL ON CLOUD INFRASTRUCTURE

CHIDAMBARAM[1], CHANDRASEKAR[2]

[1] Karpagam University, Computer Science department, Pollachi Main Road,
Eachanari Post, Coimbatore, Tamilnadu, India.
[2] Assistant Professor, Computer Science department,
Periyar University, Salem, Tamilnadu, India.

E-mail: [1]chidamkl@gmail.com, [2]ccsekar@gmail.com

ABSTRACT

Software-as-a-Service using admission control has been considered as the next generation software delivery model that shares the services among different tenants to improve the utility rate. A major feature of the admission control is that the tenant's resources are usually shared by the cloud. While the tenant's enjoy the convenience of resources being obtained from the cloud structure, they fear of undesired pressure and can become a significant barrier with respect to time. With this the QoS parameters is increasingly becoming the choice of many organizations. But, provisioning and scheduling execution of tenants of resources become a major thrust to be solved in addition to QoS parameters. To attain effective SaaS profits on single tenants, a software service model using schedule based fair queue weight (S-FQW) for dynamic admission control on cloud infrastructure is proposed in this paper. To start with, the work focus on reducing the time complexity using Schedule based Fair Queue Weight for each tenant. The Schedule based Fair Queue Weight is developed to schedule the software services based on the weighted approximated processor sharing. The weight is then assigned based on the time period the request is made by the tenant to cloud structure. Subsequently the weighted value is used to identify the positional point of the tenant to deliver software service using cloud infrastructure with higher flexibility rate. The dynamic admission control task in S-FQW model helps to derive the positions of several dynamic set of tenants. Finally, with this even the randomly distributed tenant also reaches higher profit rate using linear interpolation method providing software service at flexible rates by reducing the time complexity even on randomly distributed tenants. Various statistical parameters are measured and compared with the existing state-of-the-art works on cloud structure using Amazon web service dataset. Experiment is conducted on factors such as time complexity, total software profit rate on single tenant, average response time to tenant and software delivery rate.

Keywords: *Semi-Customized Dynamic Admission Control, Schedule Based Fair Queue Weighted, Cloud Infrastructure, Linear Interpolation, Software-As-A-Service*

1. INTRODUCTION

The significance of cloud computing has increased as it provides the clients or customers with an increased probability of savings in terms of cost. A tenant in cloud computing as referred to as a set of users share similar types of views onto an application. But any how this comes along with isolation of tenants. Due to this, software-as-a-service application has received greater attention in cloud computing. Many efforts have been undertaken to provide measures for this.

Service Measurement Index Cloud (SMICloud) [1] framework evaluated the quality and ensured mechanisms for including prioritization of cloud services by including efficient ranking mechanisms. Though ensured QoS, scheduling execution of applications remain unsolved. To overcome the limitations, SLA based Admission Control (SLA-AC) [2] was introduced

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

to enhance the rate of profit by introducing three algorithms. But the time complexity involved during the design of minimizing the number of VMs, rescheduling and attaining penalty delay was observed to be high.

With the increasing demand for computation at several levels including social interactions, businesses, education, commerce etc., the application of online banking, e-commerce, SaaS has resulted in diversities at an enormous scale. A novel allocation method called as SLA-based resource allocation [3] for multi-tier applications in cloud environment was designed using force-directed search algorithm to solve the issues related to computing and memory. But the response time was observed to be high. To solve this problem, a percentile cluster initialization [4] model was designed and was proved to be highly reliable. However, security remains unaddressed. WRF as a service [5] was designed that included federation between partners and also included measures for cost and time.

One of the most updated trends involved in distributed type of computing is cloud computing that provides both hardware models and software applications as services. The cloud users at the other end can use these services on the basis of Service Level Agreement (SLA). Though several papers have been addressed on the basis of scheduling, a novel mechanism called as the Partial Critical Paths (PCP) [6] was designed to minimize the cost involved during workflow execution using fair path scheduling algorithm. However, measures were not taken for heterogeneous workload.

The problem of resource allocation at data centers were handled in [7] that included heterogeneous workload model and ensured dynamic resource provisioning. But measures were not included for optimizing the workload. A fitness function and optimization method was introduced

in [8] on the basis of time violation for each tenant. With this scalability was ensured.

In this work, a software service model on cloud infrastructure using schedule based fair queue weight for dynamic admission control (S-FQW) has been proposed to attain SaaS profits on single tenants. The proposed system utilizes the weighted approximated processor sharing which helps to improve the flexibility rate using computed weighted value. The performance of the proposed network has been investigated using Amazon web service dataset to measure the software profit rate on tenant, time complexity by applying a fair queue weighted value.

The proposed model synergistically integrates the output obtained from weighted approximation and applies it with the queue length to obtain the positional value of the tenant that further reduces the cost factor. The performance of the proposed model is evaluated using different requests made by the tenants and performance indices.

The rest of the paper has been organized as follows. Section 2 gives a brief model for software-as-a-service and several admission control mechanisms with their briefing and limitations. Section 3 presents the proposed model with neat architecture diagram and algorithmic described included with an elaborated description of the same. Section 4 present the experimental analysis and metrics considered for the design of the model whereas Section 5 includes simulation results. Section 6 concludes the work with concluding remarks.

2. RELATED WORKS

With the emergence of cloud infrastructure, material management becomes highly complicated. Several construction patterns of cloud was deployed in [9] by designing an optimized construction model in cloud environment called as Global Authentication Register System (GARS), to minimize the material management involved during privacy and security of the cloud users. Though security was strengthened but involved time complexity and therefore increased the cost. To optimize the cost factor, Environmental Decision Support Systems [10] was designed that not only increased the decision making process but also drastically minimized the cost. An interactive queuing model was introduced in [11] that included the solution for deadlock by applying banker's algorithm. Though efficient dynamic resource allocation was ensured but security was not included for queuing model.

A fair queuing with security of low complexity was included in [12] using group multi resource round robin by providing near perfect fairness. One of the well comprehended and learnt concepts is the biological evolution that has further provided us with the proof of a routine incline for several years. In [13] Software as a Service in the Cloud was compared hitherto with the Biological evolution.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The Software-as-a-Service (SaaS) model makes a significant contribution of enabling application service providers by delivering their applications through cloud computing infrastructures. Despite, with their nature of sharing, SaaS clouds are highly susceptible to malicious intruders. IntTest [14] was presented to provide a scalable and significant service integrity attestation model SaaS. However, the model seems to be input deterministic.

A novel methodology to integrate agent technology and cloud infrastructure for assessing plant floor system was presented in [15] based on design, runtime and offline processing. However, authentication was not ensured for large volume of data. Two factor authentication systems were introduced in [16] that included confidentiality and integrity. Elastic Extension Tables (EET) in [17] for multi-tenant in software as a service model was designed for efficient management of large volume of data.

Based on the aforementioned methods stated, a novel software service model is designed in cloud infrastructure using schedule based fair queue weight for dynamic admission control in the forthcoming subsections.

3. SOFTWARE SERVICE MODEL USING SCHEDULE BASED FAIR QUEUE WEIGHT FOR DYNAMIC ADMISSION CONTROL ON INFRASTRUCUTRE

Single tenancy software as service architecture on cloud infrastructure is used to achieve higher profit rate on each tenant. The software service model using scheduled based fair queue weight is mainly used on single tenant zone at lesser time complexity rate. In the cloud infrastructure, a single tenant denotes each company with different database system. The main objective of this paper is to meet the demand of user software services even on dynamic structure.

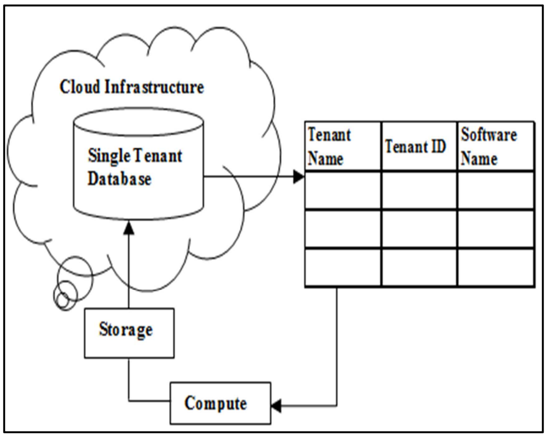

Figure 1 Single Tenant Cloud Structure

The Figure 1 illustrated above shows the structure of single tenant cloud structure. In the single tenant architecture each tenant (i.e.,) customer maintain unique software code in S-FQW model. The unique software code results with higher technical support to the customer zone with minimal time complexity rate. The rate of delivery is also improved using S-FQW model by using the weighted code system. The S-FQW model upgrades the functional level with reduced cost range. The single tenant database information produce effective software services without any overload. The overload is minimized using the schedule based fair queue weight for dynamic admission control on cloud infrastructure. The structure of software service model is illustrated in Figure 2 for identifying the modified phase depending on the tenant needs.

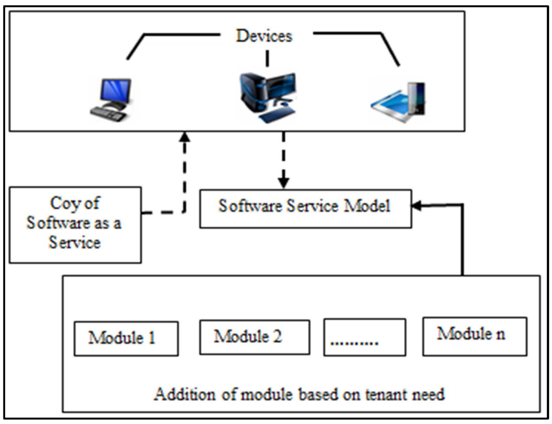

Figure 2 Structure of Software Service Model

In the S-FQW model with SaaS zone, each tenant obtains the copy of software from the cloud infrastructure and performs software service

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

depending on their needs. The software service performs minor modification of the software to meet the tenants need as described in Figure 2. In S-FQW model, it is effective as it adapt features such as additions of individual modules and channels to various database structures. The addition of the modules improves the flexibility ratio of the software and also reduces the overload factor. Architecture Diagram of S-FQW model on cloud infrastructure is illustrated in Figure 3.

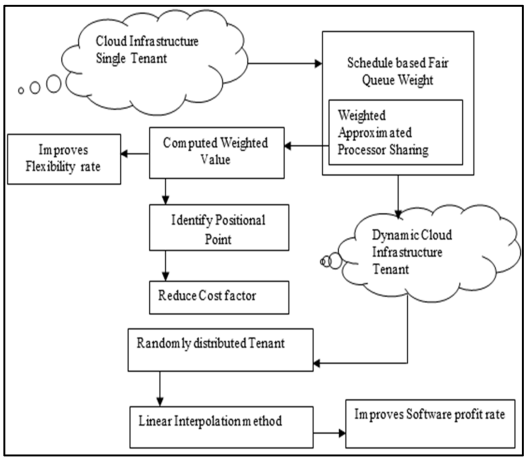

Figure 3 Architecture Diagram of S-FQWmechanism

As illustrated in Fig 3, single tenant on cloud infrastructure uses schedule based fair queue weighted method to weight the customer request based on the time period. The Weighted Approximated Processor sharing fairly accurate the request made by the tenant and produce effective computed value and improves the flexibility rate. The computed value is used for the identification of the positional value. With the obtained positional value, the cost factor of the software service on the cloud infrastructure gets reduced. The dynamic cloud infrastructure with randomly placed tenant is also worked out effectively using Linear Interpolation method. The linear interpolation method improves the software profit rate with associated weighted values.

3.1 Schedule based Fair Queue Weight

The schedule based fair queue weight (S-FQW) model initially identifies the request time from the tenant side to provide software services from the cloud zone. The request placed in the queue is used to measure the weighted vector 'weigh' of the system. 'Res' denotes the response time computed, while transferring the specific software services to the requested tenant. The schedule based fair queue weight actively controls the overload factor on cloud zone. Different types of requests are initialized from single tenant and that request is weighted using the weight factor 'w' of the tenant. Finally, the sum of all the weights from single tenant is marked as 1. (i.e., w=1).

The overload function 'v' is removed in the S-FQW model using software service model. With this removal of overload on the cloud software services reduces the time complexity rate. The schedule based fair queue weighted formularization without any overload is described as,

$$Fair\ Queue\ Weighted\ (Q) = \exp\left(\frac{Res_i(weigh_i - v_i}{v_i}\right) \quad (1)$$

The overload $'v_i'$ is removed from requested software service by single tenant cloud zone. The overload is separated from weight value to attain the response with minimal response time. The minimal response time improves the software delivery rate at the higher level. The degree of the lesser overload in S-FQW model further reduces the time complexity rate. The schedule based fair queue weighted uses the weighted approximated processor sharing to perform the summing operation on the tenant requested software's.

The approximation work is carried out in the S-FQW model by summing of the requests. The request time used to assign the weighted values and computed as,

$$Weighted\ Approximation\ (WA) = \frac{Res(weigh_i)}{(weigh_1 + weigh_2 + weigh_3 + \cdots . weigh_n)} \quad (2)$$

The weighted approximation result improves the flexibility rate by reducing the response time. By regulating the weights dynamically, the randomly placed tenants are also processed effectively. Dynamic structure is provided in brief through linear interpolation in section 3.2. The approximated weighted fairness achieves higher flexibility and identification of the positional value reduces the cost factor. The positional value of software services in S-FQW model is computed as,

$$Positional\ Value\ (P) = 1 + \exp\left(t\left[\frac{WA - Q_i}{Q_i}\right]\right) \quad (3)$$

Tenant requested on the queue are worked to identify the positional value 'P' in the cloud

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

zone. The cloud zone uses the weighted approximated result that subtracts the queue length Q_i with the weighted approximation WA and identifies the positional value. The positional value on the S-FQW model is measured exactly to identify the reduction on the cost factor. The lesser the cost factor, the higher the flexibility rate.

3.2 Linear Interpolation method

Let us assume a randomly placed single tenant 'R' on cloud infrastructure to compute the software profit rate. Random tenant places the request from 1 to 'N' on different type of software services on the queue 'Q'. The Queue takes all these requests and computes the software profit rate by avoiding time complexity on successful delivery of services. Let us consider 'T' be a tenant with random positioned services taken to perform linear interpolation on the S-FQW model. The linear interpolation for 'T' software services is computed as,

$$Q(T) = Q_{linear}\,(T) + E(T) \tag{4}$$

The linear interpolation Q_{linear} of tenant is added with the higher order estimate of error function. In order to improve the software profit rate, the S-FQW model uses higher estimates to different type of tenants. The random distributed tenant computation using linear interpolation is summarized using the below algorithmic step.

// **Linear Interpolation**
Begin
Step 1: Randomly placed tenant request on 'Q'
Step 2: Randomly moved vector points from 'x' to 'y'
Step 3: Linear interpolation $Q_{linear}\,(T)$ is formularized as
  Step 3.1: Compute $\frac{y-y_0}{x-x_0}$, where x_0 -initial position were moved to y_0
  Step 3.2: $y = \frac{x-x_0}{y_0}$ used to reach the linear queue position
Step 4: Linear position uses schedule based fair queue weight 'Q' value
Step 5: 'Q' value used for random position to attain the linear position rate
Step 6: Compute higher order estimate E (T) to improve software profit rate
  Step 6.1: Formularize $E(T) = P + higher\ (Weigh)$
  Step 6.2: Higher order Weigh from the queue improves software profit rate on summing with 'P'
End

The above algorithmic step describes the construction of linear interpolation method to order the random position. The linear form $Q_{linear}\,(T)$ is formularized with 'x' and 'y'. The software requested from 'x_0' time is moved to the 'y_0' time, then linear formation is carried out to compute the 'y' position. The exact positioning of the request time from the tenant reduces the time complexity rate. The higher order is also estimated using S-FQW model using E (T) that in turn further improves the software profit rate. The higher order estimation performs summing operation on the identified positional value with higher order weight value to achieve the E (T) value. The approximated processor sharing holds all weight factors of tenant 'T'. From that, the higher value weight function is used to compute the E (T).

4. EXPERIMENTAL EVALUATION

Software service model using schedule based fair queue weight (S-FQW) for dynamic admission control on cloud infrastructure uses the CloudSim platform to perform the experimental work. CloudSim is implemented on using Amazon web service dataset in JAVA platform with different performance factors. Cloudsim goal is to provide a global and extensible simulation framework that facilitates model, simulation, and experimental evaluation. The emerging cloud computing infrastructures and application services allow the users to focus on increasing the software profits on cloud infrastructure.

The results are investigated with the small stage information which is obtained from experimental work. A data center comprises of many software's with CPU core equivalent to 500, 1000 and 1500 Microprocessor without Interlocked Pipeline Stages (MIPS). The 4GB RAM is used for the experimental work and 1 TB of storage of single domain information. The proposed S-FQW model is compared against the existing Service Measurement Index Cloud (SMICloud) [1] framework and SLA-based Admission Control (SLA-AC) [2]. S-FQW model experiments the work on factors such as time complexity, total software profit rate on single tenant, average response time of tenant, software delivery rate.

Time complexity TC refers to the ratio of time taken by the tenant when tenant i $Tenant_i$requested a software service from cloud zone and the total number of software service SS_irequests made given as

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$TC = \sum_{i=1}^{n} \frac{Tenant_i}{SS_i} \quad (5)$$

The total software profit rate on single tenant is measured from (4) which is the software profit rate for single request. In a similar manner, N different types of software services on Queue 'Q' is formalized as given below

$$\sum_{i=1}^{N} Q(T)_i = Q_{linear}(T)_i + E(T)_i \quad (6)$$

The average response time ART of tenant is the ratio of time difference between the software service requests made by the tenant SS_{Req} and software services provided by the cloud zone SS_{Prov} to the software service requests made SS_{Req}.

$$ART = Time\left[\frac{SS_{Req} - SS_{Prov}}{SS_{Req}}\right] \quad (7)$$

Finally, the software delivery rate measures the rate at which the software as requested by the tenant from the cloud zone is made available to the tenant at minimum response time. The software delivery rate is measured in terms of percentage (%).

5. RESULTS ANALSIS OF S-FQW

The result analysis of Software service model using schedule based fair queue weight (S-FQW) for dynamic admission control on cloud infrastructure is compared existing Service Measurement Index Cloud (SMICloud) [1] framework and SLA-based Admission Control (SLA-AC) [2]. Table 1 represents the time complexity obtained using Cloudsim simulator and comparison is made with two other methods, namely SMICloud [1] and SLA-AC [2].

Table 1 Time Complexity By S-FQW, Smicloud And SLA-AC

Number of requests made by Tenant on (Q)	Time Complexity (ms)		
	S-FQW	SMICloud	SLA-AC
2	0.253	0.315	0.435
4	0.278	0.345	0.455
6	0.452	0.385	0.523
8	0.485	0.625	0.775
10	0.515	0.645	0.795
12	0.725	0.815	0.835
14	0.785	0.855	0.855

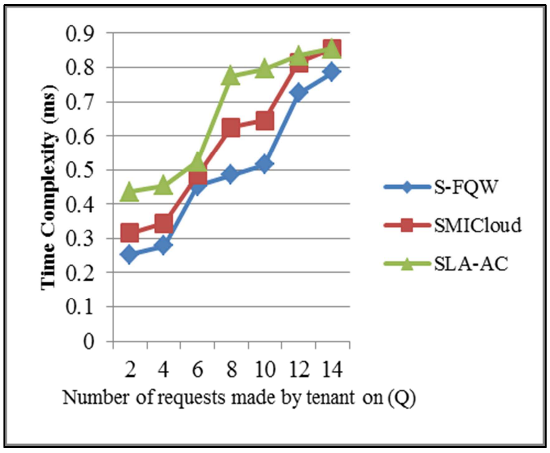

Figure 4 Measure Of Time Complexity

Figure 4 show that the proposed S-FQW model reduces the time complexity when compared to SMICloud [1] framework and SLA-AC [2]. This is because of the application of linear interpolation method in S-FQW model with varying range of requests made by the tenant based on the requirements of the tenants that provide more elaborated information of unique software code and higher technical support minimizing the time complexity by 11 – 36 % when compared to SMICloud framework. In addition to that with the use of higher order weight based on the time period the request made by the tenant minimizes the time complexity by 8 – 71 % than the SLA-AC.

Table 2 Number Of Requests Made By Tenant And Total Software Profit Rate

Number of requests made by Tenant on (Q)	Total Software Profit Rate (%)		
	S-FQW	SMICloud	SLA-AC
2	58	46	41
4	65	53	48
6	68	56	51
8	72	60	55
10	74	62	57
12	78	66	61
14	82	70	65

The comparison of total software profit rate is presented in table 2 with respect to the number of software requests made by the tenant in the ranges of 2 and 14. With the increase in the number of requests made by tenant, the total software profit rate also gets increased.

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

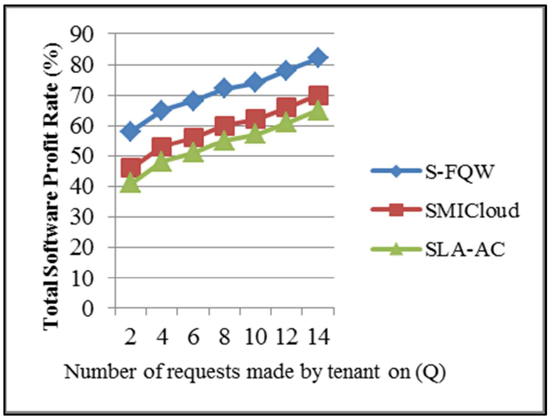

Figure 5 Measure Of Total Software Profit Rate

To perceive the performance of total software profit rate, comparison is made with two other existing works Service Measurement Index Cloud (SMICloud) [1] framework and SLA-based Admission Control (SLA-AC) [2]. In figure 5, the number of requests made by the tenant is varied between 2 and 14. From the figure it is illustrative that the total software profit rate is higher specificity is penetrating or increased using the proposed S-FQW model when compared to the state-of-the-art works. This is because with the application of weighted approximated processor sharing, the S-FQW model based on the request made by the tenant effectively produces the computed value and therefore increasing the total software profit rate by 14 – 20 % when compared to SMICloud [1] framework. Furthermore, the approximated weight achieves higher flexibility using schedule based fair queue weighted on each software service requests made by the tenant, the total software profit rate is increased using S-FQW model by 20 – 29 % than when compared to SLA-AC [2].

Table 3 The performances of average response time on S-FQW, SMICloud and SLA-AC

Number of requests made by Tenant on (Q)	Average Response Time (ms)		
	S-FQW	SMICloud	SLA-AC
2	0.222	0.303	0.401
4	0.247	0.333	0.421
6	0.421	0.373	0.492
8	0.454	0.612	0.74
10	0.484	0.632	0.762
12	0.694	0.801	0.803
14	0.754	0.841	0.832

The average response time for S-FQW model is elaborated in table 3. We consider the method with varying number of requests considering 14 for experimental purpose using Cloudsim.

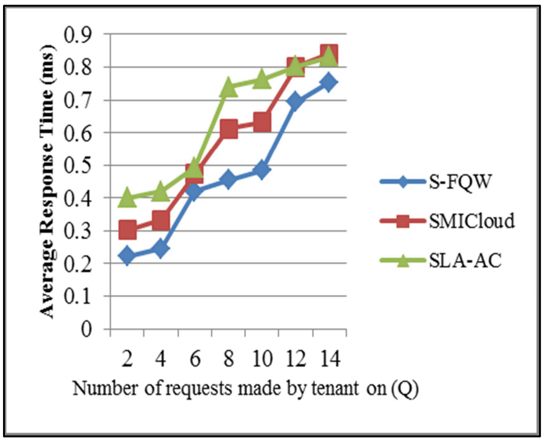

Figure 6 Measure Of Average Response Time

In figure 6, we depict the efficiency of average response time attained using software requests of size 2 to 14 for experimental purposes. From the figure, the metric average response time for the requests placed by the tenant's are minimized the proposed S-FQW model when compared to two other existing works SMICloud [1] framework and SLA-AC [2]. Besides we can also observe from the graphical representation that by increasing the number of requests, the average response time is increased using all the methods. But comparatively, it is observed to be of lower in S-FQW model because by identifying the positional point of the tenant using the weighted value with

Journal of Theoretical and Applied Information Technology
10th February 2015. Vol.72 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

higher flexibility rate the average response time gets reduced by 11 – 36 % than SMICloud framework. With the identified positional points of tenant, the dynamic admission control is effectively performed on randomly distributed tenants and therefore reducing the average response time by 10 – 80 % than compared to the SLA-AC [2].

Table 4 Tabulation For Software Delivery Rate

Size of software request made by Tenant on(MB)	Software Delivery Rate (MB)		
	S-FQW	SMICloud	SLA-AC
2000	1880	1550	1300
4000	3750	3300	2750
6000	5800	5200	4250
8000	7950	7400	6500
10000	9900	9000	8000
12000	11050	10800	9250
14000	13150	13000	12150

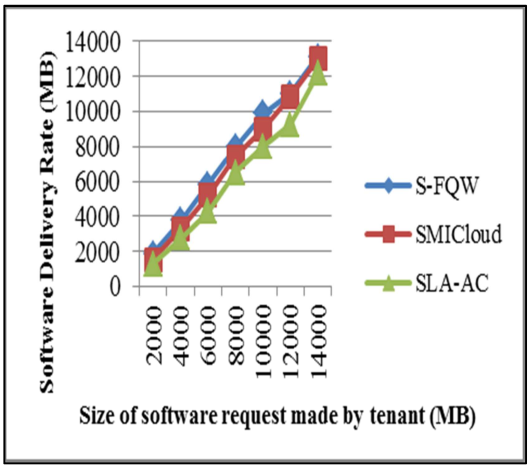

Figure 7 Measure Of Software Delivery Rate

Table 4 and Figure 7 illustrate the software delivery rate versus the size of software requests made by the tenant in terms of MB for experimental purpose conducted using Cloudsim. From the figure we can note that the software delivery rate attains 10.34 % improved for software request of size 6000 MB when compared to SMICloud [1] framework and 26.72 % improved when compared to SLA-AC [2] which shows that there is a significant gain using the proposed S-FQW model. This is because, with the application of weighted code system, the weighted approximated processor shares the software service based on time period the request made by the tenant resulting in increased software delivery rate by 1 – 17 % compared to SMICloud framework. Further, with the aid of weighted approximated processor sharing the overload is separated from weight value and increasing the software delivery rate by 7 – 30 % compared to SLA-AC [2].

6. CONCLUSION

In this paper, we studied an effective software service model to attain effective SaaS profits on single tenants using schedule based fair queue weight for dynamic admission control on cloud infrastructure. The goal of our software service model is to improve the software delivery rate by applying weighted code system which significantly contributes to the relevance. To do this, we first devised a weighting measure called as the fair queue weight to determine the software service as requested by the tenant at a specific time. Then, based on this measure, the overload function is removed followed by which an approximation work is carried out which reflects the response time of tenants. In addition, the positional values of the software services are evaluated based on the queue length and weighted approximation value. Moreover, to improve the software profit rate for randomly placed single tenant, linear interpolation is applied for different tenants using Linear Interpolation algorithm based on the software service requests made by the tenants on cloud infrastructure. Through the experiments using Amazon web service datasets, we observed that our software service model provided improved software delivery rate compared to existing state-of-the-art works. In addition, our linear interpolation algorithm effectively reduced the average response time of tenant and even improved the total software profit rate on tenants'.

REFERENCES

[1] Saurabh Kumar Garg, Steve Versteeg, Rajkumar Buyyaa," A framework for ranking of cloud computing services", Future Generation Computer Systems, Elsevier journal., 2012
[2] Linlin Wu., Saurabh Kumar Garg., Rajkumar Buyya., "SLA-based admission control for a Software-as-a-Service provider in Cloud

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

computing environments, Journal of Computer and System Sciences., Elsevier journal., 2012

[3] Hadi Goudarzi and Massoud Pedram," Multi-dimensional SLA-based Resource Allocation for Multi-tier Cloud Computing Sysems" IEEE, 4th International Conference on Cloud '11 Pages 324-331, Oct 2011

[4] Kaiqi Xiong and Sang Suh," Resource Provisioning in SLA-based Cluster Computing", Springer, pp. 1 – 15, Job Scheduling Strategies for Parallel Processing, Vol. 6253, Mar 2010

[5] David Villegas, Norman Bobroff, Ivan Roderob, Javier Delgado Yanbin Liu, Aditya Devarakonda, Liana Fong, S. Masoud Sadjadi, Manish Parashar," Cloud federation in a layered service model", Journal of Computer and System Sciences, Elsevier, May 2011

[6] S.Abrishami, M. Naghibzadeh," Deadline-constrained workflow scheduling in software as a service Cloud", Scientia Iranica, Elsevier, May 2012

[7] SaurabhKumarGarg, AdelNadjaranToosi, SrinivasaK.Gopalaiyengar, RajkumarBuyya," SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter", Journal of Network and Computer Applications, Elsevier, Aug 2014

[8] Rouven Krebs, Philipp Schneider, Nikolas Herbst," Optimization Method for Request Admission Control to Guarantee Performance Isolation", ACM, Proceedings of the 2nd International Workshop on Hot Topics in Cloud Services, Mar 2014

[9] Chih-Yung Chen and Jih-Fu Tu," A Novel Cloud Computing Algorithm of Security and Privacy", Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2013

[10] Alexander Sun," Enabling collaborative decision-making in watershed management using cloud-computing services", Environmental Modelling & Software, Elsevier, Nov 2012

[11] Sandeep K. Sood," Dynamic Resource Provisioning in Cloud based on Queuing Model", International Journal of Cloud Computing and Services Science (IJ-CLOSER) Vol.2, No.4, August 2013, pp. 314~320

[12] Wei Wang, Ben Liang, Baochun Li," Low Complexity Multi-Resource Fair Queueing with Bounded Delay", IEEE Conference on Computer Communications, Feb 2014

[13] Atif Farid Mohammada, Joumana Darghamb , Hamid Mcheickc, Attia T. Noora," Software Evolution as SaaS: Evolution of Intelligent Design in Cloud", Elsevier, The 4th International Conference on Ambient Systems, Networks and Technologies (ANT-2013)

[14] Juan Du, Daniel J. Dean, IEEE Yongmin Tan, Xiaohui Gu, Ting Yu," Scalable Distributed Service Integrity Attestation for Software-as-a-Service Clouds", IEEE Transactions on Parallel and Distributed Systems, Volume.25, Issue.3, Pages 730-739, Mar 2014

[15] Francisco P.Maturana, Juan L. Asenjo, Neethu S. Philip, and Shweta Chatrola," Merging Agents and Cloud Services in Industrial Applications", Hindawi Publishing Corporation Applied Computational Intelligence and Soft Computing Volume 2014

[16] Jin-Mook Kim and Jeong-KyungMoon," Secure Authentication System for Hybrid Cloud Service in Mobile Communication Environments", Hindawi Publishing Corporation International Journal of Distributed Sensor Networks Volume 2014

[17] Haitham Yaisha,b, Madhu Goyala,b, George Feuerlicht b," Evaluating the Performance of Multi-tenant Elastic Extension Tables", Elsevier, ICCS 2014. 14th International Conference on Computational Science May 2014