Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/110318

Agent-Based Resource Management In A Cloud Environment

Seenuvasan P.[1,*], *Kannan A.*[2] *and Varalakshmi P.*[3]

[1] Department of Information Technology, University College of Engineering, Villupuram, Tamilnadu, India
[2] Department of Information Science and Technology, Anna University, Chennai, Tamilnadu, India
[3] Department of Computer Technology, Madras Institute of Technology, Chennai, Tamilnadu, India

Received: 23 Dec. 2016, Revised: 10 Mar. 2017, Accepted: 15 Mar. 2017
Published online: 1 May 2017

Abstract: An appropriate load balancing mechanism is necessary for storage and retrieval of data using among the resources in a cloud data center to improve the efficiency. In order to balance the load, the resources need to be monitored and both the parties namely cloud customer and cloud service provider should negotiate and sign up with a Service Level Agreement (SLA) to keep smooth relationship. Hence a new Precedence Based Monitoring (PBM) algorithm is proposed in this paper for monitoring the cloud resources based on time, event and precedence of the resources.Moreover a new technique for Negotiation and SLA formation between the lead node of cloud site and consumer are proposed with intermediary agents. Both the consumer and the cloud provider are benefited by this proposed Reduced Penalty Class Algorithm (RPCA). If the user requested lease is not able to be provisioned on a particular cloud, then the intermediary agent of current cloud migrates the request of the customer to the intermediary agent of the neighboring cloud. Therefore a new mechanism for centralized load balancing among the computing nodes in a cloud site and decentralized load balancing across the cloud sites is proposed in this paper. The results obtained from the proposed system shows improvement in user satisfaction level and resource utilization of cloud provider that is the reduced due to the effective sharing of resources and efficient load balancing.

Keywords: Precedence Based Monitoring algorithm, Reduced Penalty Class Algorithm, Service Level Agreement, Intermediary agents, Hybrid Load balancing.

1 Introduction

Hardware and software are provided as on-demand services to the booming Cloud computing Technology. This technology not only supports IT industries, but also the individual users. Before provisioning the service from any cloud providers, users should negotiate with the cloud providers just by submitting their demands, which is the first step in the Service Level Agreement (SLA) [15,44] formation. Moreover an SLA brief about the agreed-upon demands and services, assurances, and remedial actions in case of violations among the cloud providers and cloud users. If the cloud providers agreed upon their demands, an SLA will be formed. Negotiation may be performed directly between the users and the cloud providers, or through intermediary agents. If a user requires more than one type of service, he may be supported through a number of agents for each type of services.

Monitoring[14,22] is one of the main issues in the cloud. Moreover a continuous monitoring of the current status of the resources on the cloud is necessary to improve the efficiency of the cloud providers to provide a better quality service to the cloud users. The status of cloud sites is monitored periodically or based on threshold or certain events.

Load balancing[6,7,13,18,23] is a challenging issue of cloud computing. Centralized and decentralized are the two basic types of load balancing algorithms available. Only one node will act as the centralized controller in the cloud data center in a centralized load balancing approach. This node only allocates the users requests to each of the member nodes. The member nodes only execute the requests assigned by the centralized controller. Hence the centralized controller is overloaded at many situations which leads to single point failure. More resources in the cloud data center are involved in the decentralized load balancing approach in making the load balancing decisions. This is because the decentralized algorithm does not have a single point

* Corresponding author e-mail: psvasanucev@gmail.com

failure problem and it not only improves the scalability but also provides better fault tolerance capacity, However it increases the communication overhead.

In this paper, the negotiation process takes place with the help of an intermediary agent (third party) and then the SLA is signed. A new scheduling mechanism is proposed, which is used to classify the leases (i.e. user requests) into classes and servicing them in such a way that the penalties of both the cloud provider and the customer got reduced. Both periodic and a threshold based monitoring methods are used in this proposed work according to the conditions present in the cloud site. Priority or the precedence is also considered for the proposed monitoring system. A centralized load balancing mechanism in the intra cluster sites and decentralized load balancing method across the inter cloud sites are proposed in this paper.

The remainder of this paper is organized as follows: In Section 2, the related work is discussed. Section 3 presents the architecture of the proposed work. Section 4 describes the proposed PBM algorithm, lease request generation, negotiation, SLA formation and RPC algorithm. The experimental results are explored in section 5. In section 6, Conclusion and Future work are provided.

2 Related Work

Cloud computing offers on-demand services to customers. In order to avoid discrepancies, negotiation and signing of SLA at the initial stage is necessary. Monitoring the cloud resources and load balancing among the resources are the main issues need to be considered to improve the efficiency of cloud sites. Many works on cloud scheduling, load balancing and resource provisioning are available in the literature.

Thomas Rings et al. [26], proposed the integration of grid and cloud methodology standards into Next Generation Networks. Roger Halbheer et al. [17], discussed about the fundamental challenges and benefits of cloud security. Kai Hwang et al. [10], proposed a reputation system in which diverse security procedures are recommended to guard cloud resources. Jian Wang et al. [8], proposed new privacy preserving technologies in their paper. Weili Huang et al. [28], provided a the comparison between the firewalls in cloud data centers and the traditional data centers.

Sumathi. G et al. [22], proposed nw technique for the selection of time period in order to monitor the resources at Grid site. The issue with their model is that if the time interval considered is not optimal the number of announcements to be made will be large in number. To rectify this, a solution with dynamic time interval is proposed in this paper. Min Li and Yisheng Zhang [14], proposed High Performance clusters for monitoring the current condition of all the resources. Xioojiang et al. [30], suggested to employ a local manager for distributed monitoring approach in each of the clusters. The local manager is accountable for monitoring all the resources in the cluster. Manvi et al. [12], suggested to assign priority to the resources and based on the importance of the resources, frequency of the monitoring is decided based on the priority in wireless grid. Wu-Chun Chung et al. [29], 2009, discussed about the monitoring of grid resources based on time and change events.

Mohammed Alhamad et al. [15], briefed about the SLA parameters like CPU speed, memory required, required software, and etc. for different type of cloud services. Keerthana Boloor et al. [11] discussed about the reduction of penalties in case SLA violation. But, Reduced Penalty Class Algorithm (RPCA) is proposed by the authors to benefit both the users and the providers . Ivona Brandic et al.[5] discussed about the SLA based cloud architecture and the mitigation in case of violation of the SLA. Hien Nguyen Van et al.[2] proposed an automated resource manager to optimize the total service function. Vincent C Emeakaroha et al. [27] proposed a communication framework to increase the scalability. Sebastian Hudert et al. [19] proposed a bilateral and multilateral negotiation in two stages.

Shirlei Aparecida de Chaves et al. [20], discussed about choosing a best provider through a broker by considering various criteria. Hima Prasad et al.[9] proposed SLA formation for huge data processing services and provisioning the resources based on parameters like network speed, reliability and throughput. Stefano Ferretti et al. [21] proposed a Sec-SLA framework which is nothing but SLA formation based on security parameters. Mohd Farhan Md Fudzee et al. [16] proposed a cloud architecture, in which monitoring and scheduling are done with a module of Virual Enecuion Environment manager and load balancing is done with the service manager module. New load balancing algorithm is proposed by Ruchir Shah et al. [18], for Grid in which buddy set is used for each of the processes. The issue with the above paper is that buddy set will not be updated regularly. Thus the load balancing is done with old data. This is rectified with the proposed PBM algorithm at regular intervals.

Belabbas Yagoubi et al. [1], discussed link utilization based Load Balancing framework for Grid. Jacob Honore Broberg et al. [6], proposed a solution for balancing the load using grid manager in a tree structured grid environment. Martin Randles et al. [13], said that and Honey bee algorithm has given a better performance for balancing the load of cloud resources., Job combinations and assignment methodology for load balancing is proposed by Hua-Feng Deng et al. [3]. Iman Barazandah et al. [4], proposed a comparative study on two biased load balancing algorithms namely, dynamic biasing algorithm and minimum load state round robin algorithm in distributed systems. Janhavi et al. [7], made a comparison of different load balancing algorithms for Grid. Suri. P. K. et al. [23], proposed proximity based load balancing algorithm for the Grid.

Yi Zhao and Wenlong Huang [31], proposed a Virtual Machine(VM) migration based load balancing algorithm for Cloud. Takahiro Hirofuchi et al. [24], proposed relocatable VM Services on Clouds. A new VM can be created as the leases arrive and it can be deleted when its usage is over. Chieu et a. [25], improved the scalability of Web based Applications in Cloud environment. In the proposed work, when leases arrive and only if the load minimization criterion is satisfied, a VM is created for the execution of the lease. Jorge E. Pezoa et al. [32], developed a probabilistic model which takes decisions on the load balancing action only after the nodes received the message in harsh environment without considering the knowledge of SLA. Elena Renda M et al. [33], presented a methodology which applies different methods for load balancing even, if the uniformity assumptions, such as the location of nodes and the query sources were filled or not. The approach did not suggest any improvement in system performance related to Quality of Service(QoS). Hung-Chang Hsiao et al. [34], proposed a load re-balancing algorithm in MapReduce based applications as well as to achieve a fast convergence rate in cluster environment without overloading the nodes. Though it reduces the network traffic and avoids the dependency of central node, QoS parameters were not considered.

Dinil Mon Divakaran et al. [35], suggested a three phase Integrated Resource Allocator (IRA) for grouping, discovering and finally allocating the resources for the accepted request to achieve a guaranteed performance for the tenants in multi-tenant data centers without any SLA negotiation. Dario Bruneo et al. [36], endorsed a Non-Markovian Stochastic Petri net model to improve performance factors, by implementing preprocessing, mapping and composition techniques without SLA violation. Hossein Morshedlou et al. [37], proposed a method to decide and release the Virtual Machine (VM) resources for the user request with the help of User Broker and VM Broker by taking into account the characteristics of users, like willingness to pay. It considers only user satisfaction level as a vital factor, it failed to exploit the penalty criteria for the users.

Mario Macias and Jordi Guitart [38] discussed about the formation of SLA in business perspective with two sets of policies such as classification of clients and Revenue Maximization. Selection of potential clients to maximize the revenue is important to cloud providers, which are considered in SLA formation. Apart from the maximization of resource utilization and profit at cloud sites, Saurabh Kumar Garg et al. [39] suggested to ensure the QoS in the specification of SLA formation. Federated cloud network across different Software Defined Network with related SLA are briefed by Alexander Stanik et al. [40]. High job success rate is obtained through the proposed agent based discovery and negotiation framework using the contract net protocol and acquaintance networks in agents is discussed by Kwang mong Sim et al. [41]. Secured Temporal Log Management Techniques and Intelligent Temporal Role Based Access Control for Data Storage in cloud database are discussed in [42] and [43]. Energy-Efficient Server-Consolidation Based Resource Allocation in Cloud environment is dicussed in [44].

In the proposed Precedance Based Monitoring (PBM) algorithm for monitoring purposes, time and event based monitoring are used along with considering the precedence of the resources also. Both the consumer and the cloud provider are benefited by this proposed Reduced Penalty Class Algorithm (RPCA) with formed SLA. In RPCA, aggregate score of the each of the lease is calculated based on the weight given by the user to each of the required resources. By considering the aggregate score of the leases, the leases are placed in the appropriate class (queue). The classes are prioritized based on aggregate class score. The leases will run for a trail period. Then the class which is having the highest penalty will be given higher priority. The lease, which is having a higher penalty for each of the classes will be scheduled first. So both customer and service provider will be benefited by the proposed RPC algorithm with reduced penalty.

3 Proposed Framework

Monitoring, negotiation, SLA formation and load balancing issues are mainly considered in this work to increase the performance of cloud resources. A solution covering all these above stated areas is concentrated in order to provide a Dual Party Beneficiality beneficial to both the cloud provider and the consumer. The following components are considered in this proposed work:

–User node which sends the lease request.
–Lease (request) is the service that the user wants from the cloud.
–Intermediary agent is an agent between the user and the lead node of cloud sites for the negotiation and SLA formation.
–The lead node is present in every cloud site for monitoring and load balancing.
–Many computing nodes are present in each cloud for providing the services requested by the users.

Making use of these components, a solution is proposed containing the following steps:

–The nodes in cloud site are monitored by the lead node.
–Lease request is created from the user nodes.
–Negotiation process has taken place between the user node and the lead node of cloud sites via the intermediary agents.
–SLA is formed between the lead node and the user node.
–Scheduling the user leases using classes.
–The load is balanced in the cloud across the lead nodes through the intermediary agents.

Fig 1 shows the cloud setup of the proposed work. Every cloud sites comprises of one lead node and the corresponding compute nodes. Current status, such as current load, processing speed, available memory capacity and etc. for each of the computing nodes is monitored by the lead node, and the lead node stored this monitored status information.

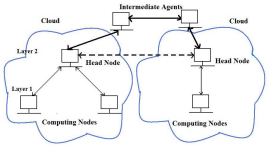

Fig. 1: Cloud Computing setup for the proposed work

4 Workflow of The Proposed Solution

Fig 2 shows the architecture of the proposed system. The proposed solutions related to monitoring, negotiation, SLA formation and load balancing are explained here.

4.1 Lease Generation, Negotiation and SLA Formation

Under this section, the lease request generation, negotiation and SLA formation are discussed.

4.1.1 Generation of Lease Request

The lease is nothing but the user request. Lease comprised of the factors like CPU speed in MHz (req_cpu), required amount of memory in Kb(req_mem), number of machines needed (req_nom), maximum waiting time for the users in seconds (req_mx_wt), deadline (req_dead), operating system required (req_os) and software required (req_soft). Based on the importance given by the users for each of the lease parameters, weights are assigned like wt_mem for representing the weights given to the required amount of memory, wt_cpu to represent the weights assigned for CPU speed, wt_wt to represent the weights assigned to the waiting time, etc. All these parameters are associated with penalties in case of SLA violation. If a provider is unable to provision the CPU speed agreed in SLA, the provider should pay a penalty to the customer and is represented as pty_cpu. If the provider is unable to provision the required amount of memory agreed in the SLA, the provider should pay the penalty to the customer and is represented as pty_mem. If the provider is unable to support the waiting time agreed in the SLA, the provider should pay the penalty to the customer and is represented as pty_wt. The user specified weight and the penalty associated with various parameters are also present in the lease request as well as in SLA.

4.1.2 Negotiation

The intermediary agent is present in each of the cloud site in the proposed work to assist the negotiation process among the cloud providers and cloud customers, which is shown in Fig 2. The user submits the request to the one of the cloud providers through the provider's intermediary agent. The intermediary agent retrieves the current resource status through the continuous monitoring process present in the lead node of the cloud site and it will check whether the requested resources are available in the cloud site. If the resources are sufficient to convince the user request, the intermediary agent informs the end of the negotiation process to both cloud user and cloud provider.

If the factors mentioned in the user lease are not convinced by the current cloud site, the intermediary agent will inform to the user about the availability of the resources in the current cloud site. If the user convinced, then the intermediary agent ends up the negotiation process and informs about this to both the cloud provider and cloud customer. If the user is not convinced, the intermediary agent migrates the user lease to the intermediary agent of the one of the nearby cloud. Then that intermediary agent will start the negotiation process. So, there is decentralized load balancing happened across the cloud sites through their respective intermediary agents. Thus, the negotiation process among the lead node and the cloud customer is fruitfully ended through the intermediary agent.

4.1.3 Formation of SLA

After the fruitful completion of the negotiation process among the cloud provider and the cloud customer, SLA is made. This agreement form includes the promises about the services going to be provided by the cloud provider and the promises made by the cloud user for provisioning of the services. The cloud provider agreed to provide the services with the parameters like CPU speed, memory capacity, operating system, software, and etc. which is requested by the cloud user. The cloud user also agreed to the payment (req_pay) specified by the cloud provider for provisioning the services. The payment to the provider is computed based on the amount of service the user is going to use over the duration. If there is any violation of these agreed properties in SLA, which is monitored by the intermediary agent, penalties from the cloud user if there is a lack of payment to a cloud provider (pty_pay) and penalties from the cloud provider if there is a lack of resources provisioning like CPU speed (pty_cpu), number

of machines (pty_nom), memory capacity (pty_mem), waiting time committed to the user (pty_wt) to the cloud user are also mentioned in that SLA. The violating of SLA properties from both the cloud user and cloud provider are taken care by the intermediary agent present in the cloud site.

4.2 Precedance-based Monitoring (PDM) Algorithm

The lead node continuously monitors the status of the resources present in the cloud site based on two conditions. One is event based monitoring, that is whenever the Change in the capacity of the resources in successive Announcement (AC) is more than or equal to a dynamic maximum threshold value (d_mx_th), the updating of resource capacity is pushed from the resource to the lead node. The initial threshold value is set at 50% of the available capacity of the each of the resource types. Later, dynamic maximum threshold for the capacity change of each of the resource type is updated dynamically once in a time window period. It is calculated using (1) as follows.

$$d\_mx\_th = \frac{1}{NA}\left[\sum_{i=1}^{NA} AC_i\right] \quad (1)$$

Here, ACi is the ith Change in the capacity of the resource in successive Announcement and NA is the number of times the change occurs during the time window. There is a minimum threshold (min_th) value for each of the resources which is considered as 20% of the available capacity of the resources. Another condition is based on time. Notification about the change in the capacity of the resource is periodically sent to the lead node of the cloud site even though the change in the capacity of the resource is not more than or equal to the d_mx_th value but the change in capacity should be more than the min_th value. If a change in the capacity of the resources generates a greater effect on the load of a node is considered as a mission critical one. The precedence of the resource is considered for resource monitoring. Higher precedence value is given to the mission critical nodes and lower precedence is assigned to the non-critical resources. Even though the timer is not expired and change in capacity is not more than or equal to the d_mx_th but a change in the capacity is more than the min_th, the change in capacity of the resources from higher precedence node is notified to the lead node immediately. Range of values 1 to 3 are considered as higher precedence and 4 to 10 are considered as lower precedence in the proposed work. The proposed Precedence Based Monitoring (PBDM) algorithm, is explained below.

Precedence Based Monitoring (PBM) algorithm:

Let ACi is the change in the capacity of the resource in successive announcements of the resource i, min_thi and d_mx_thi denote the minimum threshold and the dynamic maximum threshold of the change in capacity of the ith resource. d_mx__thi is assumed to be 50

At the initialization, announced the current status value of each of the resources to the lead node; 30 seconds are assumed for periodic timer in this implementation; Time window for fixing the dynamic maximum threshold value is assumed to be 5 hours. Let the number of announcements in the resource i be NAi = 0; The steps of the algorithm are as follows:

```
WHILE (TRUE)
{
    IF
        the change in the capacity of the resource > d_mx_thi
    THEN
{
Notify the current status of the resource to lead node;
        NAi = NAi + 1;
Compute the new d_mx_thi;  Reset Timeri; }
}
    IF
        the timeri expires AND the change in the capacity of the resource > min_thi
    THEN
{
    Notify the current status of the resource to lead node; Reset Timeri;
    }
    IF
        the change in the capacity of the resource  <  d_mx_thi AND > min_thi AND
        timeri has not expired
    THEN
    {
```

```
    IF
        Precedencei of the current node i is < 4
    THEN
{
        Precedence is high. Announce the current change in capacity to the lead node;
        Reset Timeri;
    }
    ELSE
{
        Lower Precedence and hence this change is not announced.
    }
}
```

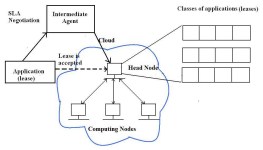

Fig. 2: Architecture of Negotiation, SLA formation and RPCA

This algorithm monitors the load more effectively.

4.3 Reduced Penalty Class Algorithm and Load Balancing in a Cloud

Under this section, the classification of leases into classes, working of Reduced Penality Algorithm Algorithm (RPCA) and load balancing are discussed. Normally the leases are scheduled into the appropriate cloud based on the functional requirement of the lease. Here, the score of the lease is calculated based on the weight of individual resources present in computing nodes. The Weight of individual resources is specified by the user based on the importance of the resources the user considered.

4.3.1 Reduced Penalty Class Algorithm

In the proposed algorithm called Reduced Penality Class Algorithm(RPAC) Leases are scheduled into different classes of queues. Different classes of queues are assigned with different class_scores and the accepted leases are also assigned with aggregate_score (aggr_score). The aggregate_score of each lease is computed using (2) as follows.

$$aggr\_score = (req\_mem^{*}wt\_mem) + (req\_cpu^{*}wt\_cpu) + (req\_max\_wt^{*}wt\_wt) + (req\_pay^{*}wt\_pay) \quad (2)$$

RPC algorithm:

```
Let li denote the ith lease which arrives at cloudp.
```

Five classes of queues are considered here where the aggregate_score of each lease with 1 to 2 are placed in one class of queue, 3 to 4 into another class of queue, 5 to 6 into another class of queue, 7 to 8 into another class of queue and 9 to 10 into another class of queue. Leases are placed into the respective queues based on the aggr_score of the lease.

The average of aggregate_score of all the leases in each class is considered as class score of that class of queue (c_score). The class with the uppermost c_score is selected and the lease with the uppermost aggr_score in that class is selected for execution. Like this, the user leases are scheduled based on their aggr_score. All the leases in all the five classes are executed for a sample period of time initially.

After the sample period of execution of the leases for all the classes of queues, the aggregate penalty of each lease (aggr_pty) is computed which is nothing but the sum of the penalties given by the customer in SLA which includes pty_mem, pty_cpu, pty_wt and pty_pay. The class penalty (class_pty) is just the aggr_pty of all the leases present in that class queue. Then, the class which is having the uppermost class_pty is selected and the lease with the uppermost aggr_pty in that class is selected and scheduled for the next complete execution. This way all the leases in all the class queues will be scheduled and executed. So the penalty paid by both the cloud provider and cloud user is considered and minimized in this proposed by scheduling the leases first which is having the uppermost aggr_pty from the class queue which is having the peak class_pty. Cloud provider's satisfaction and resource utilization of the cloud site is improved by reducing the penalty of users. Cloud user's satisfaction and Quality of the Service (like reduced waiting time) is improved by reducing the penalty of cloud providers. This way, both the parties like cloud user and cloud provider are benefited by the proposed work. The proposed Reduce Penalty Class algorithm is given below.

```
Let cloudp denote the pth cloud considered among the n available clouds and
let IAp denote the Intermediate-agent of cloudp.

Let req_memi, req_cpui, req_max_wti, req_payi represent the required
memory, required CPU speed, maximum waiting time and payment required for
lease li as stated in SLA.

Let wt_wti, wt_memi, wt_cpui and wt_payi denote the weights assigned
to each parameter based on the importance given for waiting time, memory,
CPU speed and user payment for lease li.

Let class Cj denote the class having highest class penalty and each class
has a queue of leases waiting to be serviced.

Let pty_wti ,pty_memi and pty_cpui denotes the penalty values for lease li.

Let aggr_ptyi represent the aggregate penalty for each lease li and aggr_scorei
is used to categorize the leases among the classes.

Let class_ptyj denote the aggregate penalty of all the leases in class Cj.

WHEN li arrives at cloudp,
IAp helps in negotiations between the lead node of cloudk and lease li
    considering all the required parameters
IF negotiation is successful between the lead node of cloudp and user
IAp helps in SLA formation between cloudp and user of lease li
Find aggr_scorei, aggr_ptyi and put li in the appropriate class
Find the class having maximum class_pty
IF class Cj has maximum class_ptyj
THEN Find the lease having maximum aggr_pty in class Cj
IF lease lk has maximum aggr_ptyk in class Cj
THEN  lease lk is chosen
Choose the computing nodes having the OS and software specified
Among these chosen computing nodes, choose the under loaded computing node
    in cloudp having free memory >= req_memk and speed >= req_cpuk
Execute the lease lk in computing node
IF waiting time > req_wt_maxk
THEN penalty pty_wtk is added to cloudp
                        IF free memory of compute node a <= req_memk
THEN penalty pty_memk is added to cloudp
                        IF  cpu speed of compute node a <= req_cpuk
THEN penalty pty_cpuk is added to cloudp
ELSE
User lease li requirements are not satisfied by cloudp
Lease li is passed from IAp to its neighbor IAq which is the
intermediate-agent of cloudq
IAq helps in the negotiation and SLA formation between lease
li and lead node of cloudq
    Then the lease li is executed in cloudq following with the above steps
```

4.3.2 Load Balancing in Proposed Work

For each cloud, the average of the normalized load of cloud, NLCavg is found by the lead node based on each of the computing node present in the cloud using (3)as.

$$NLC_{avg} = \frac{\sum_{k \in cloud} \left[S_k^* L_k(T) \right]}{\sum_{k \in cloud} S_k} \quad (3)$$

Here, the speed of each computing node k in a cloud is represented by Sk and Lk denotes the load of the computing node k. A node is said to be under loaded if its load is lesser than the NLCavg of that cloud. The number of VMs that can be instantiated in a node is basically considered as load. Using the load, the number of VMs required by the lease as specified in SLA can be created.

When the lease is allocated by the lead node to the computing node which satisfies the user requirements, the load of that computing node is also considered. Only if the load of the computing node is less than NLCavg, the lease is serviced in that computing node. If more than one compute node satisfies the user requirements, then any one of the under loaded compute node among them is chosen. Thus the centralized load balancing method is used to balance the load in each cloud site. The centralized node is the lead node which takes care of the load balancing among the computing nodes. If negotiation is unsuccessful, decentralized load balancing is considered among the near-by cloud sites through their neighboring intermediate-agents.

5 Experimental Analysis

The proposed work is implemented with Open source cloud tool Open Nebula with Xen hypervisor to setup the cloud. Initially a ssh password-less connection is established between the lead node and the computing nodes. This is necessary for the lead node to control the resources of the computing nodes efficiently. This creates a centralized communication between the lead node and the computing nodes in the cloud. The lead node of each cloud monitors the resource changes in each computing node in that cloud. The lead node stores this information as a file (oned.log). Both time based and event based monitoring is considered along with the precedence of resources in the PBM algorithm. The PBM algorithm is embedded in the Information Manager code in Open Nebula. The computing nodes id along with its change in memory at different time intervals is shown in Table 1. Here the timer is set to 30 seconds and the initial d_mx_th is set to 50 % of the capacity of resource in the computing node. The d_mx_th value gets changed based on the resource change in that computing node.

Table 1: Experimental Analysis of Precedence Based Monitoring (PBDM) Algorithm

Computing Node_id	Available free memory (kb)	Incident time of the change (seconds)
1	538624	0
1	538540	40
1	1024	60

5.1 Performance analysis for Monitoring

The update_rate and missed_update_rate are considered for estimating the performance of the proposed PBM algorithm.

5.1.1 Update_Rate

All updates required to be transmitted from the computing node to the lead node to reflect the changes instantly. The update_rate for the duration of monitoring is calculated using (4).

$$UR = \frac{N_{updates}}{D} \quad (4)$$

Here UR is update_rate, Nupdates is number of updates and D is duration of monitoring. A bigger value of update_rate consumes greater bandwidth, but the resource status is more exact. The lesser value of update_rate means consumes lesser bandwidth, but gives only lesser accurate value to reflect the status of the resources.

5.1.2 Missed_Update_Rate

To evaluate the timeliness of data update, the missed update rate of various data delivery protocols are evaluated. The missed_update_rate is nothing but the change in the status of the resource above min_th is not reflected in the lead node of the cloud and its computation is given as follows using (5).

$$MUR = \frac{N_{missed-updates}}{D} \quad (5)$$

Here MUR represents missed_update_rate, number of missed updates is denoted as Nmissed_updates, and D denotes the duration of monitoring. MUR will be very large if the min_th value is very large, that means change in resource status is not reflected in the lead node timely. So, missed_upate_rate should be less for a better performance.

The proposed Precedence Based Monitoring (PBM) algorithm is compared with the existing Announcing with Change and Time Consideration (ACTC) algorithm for these two performance measures such as update_rate and

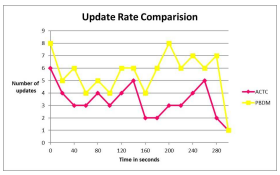

Fig. 3: Comparison of ACTC and PBM algorithms for update rate

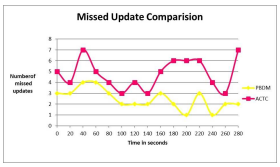

Fig. 4: Comparison of PBM and ACTC algorithms for the missed update rate

missed_update_rate for the simulation duration of 300 seconds and is shown in Fig 4 and 5 respectively.

From Fig 4, the update_rate is higher for the proposed PBM algorithm compared to ACTC at any time instant. From Fig 5, the missed_update_rate is lower for the proposed PBM algorithm compared to ACTC at any time instant. These results show the improvement of the proposed PBM compared to ACTC by reflecting needed changes in resource status more precisely to the lead node of the cloud.

5.2 Performance analysis of Reduced Penalty Class Algorithm (RPCA)

Performance measure, the penalty due to violation of SLA terms is discussed below.

5.2.1 Violation of Waiting Time

The proposed RPC algorithm with the SLA is compared with the existing gi-FIFO algorithm without SLA based on the penalty paid by the cloud provider due to the violation in waiting time defined in the SLA, which is shown in Fig 6. If the waiting time of the lease exceeds the value specified in the SLA, then the appropriate penalty must be paid by the cloud provider. Because of the reduction in providers penalty by the proposed RPC algorithm compared to existing gi-FIFO algorithm without SLA, the consumers satisfaction level is much improved by the proposed method.

5.2.2 Memory and CPU speed violation

The Fig 7 and Fig 8 show the comparison of the proposed RPC algorithm with SLA formation and without SLA formation based on the penalty paid by provider due to the violation in memory and CPU speed respectively agreed in the SLA. The result shows that the proposed RPC with SLA performs better than without SLA in terms of penalty paid by the provider.

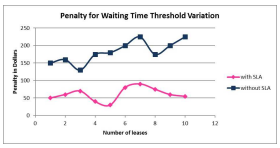

Fig. 5: Comparison for penalty paid due to the violation in waiting time

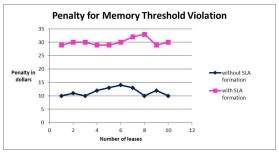

Fig. 6: Comparison of the penalty paid due to memory violation

The aggregate penalty comparison is shown in Fig 9, with and without negotiation and SLA formation. The results in Fig 6, 7 8 and 9 clearly show that the penalty is less when SLA negotiation is carried on, as the number of leases increase.

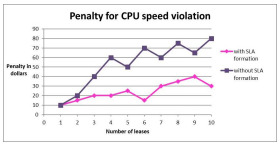

Fig. 7: Comparison of penalty paid due to violation in CPU speed

5.3 Experimental Analysis of Reduced Penalty Class Algorithm (RPCA) and Load Balancing

The proposed system considers 100 leases allocated to these two clouds, in real-time. Load balancing and

Fig. 8: Comparison of aggregate penalty with and without SLA formation

monitoring are done by the lead nodes implemented with Open nebula. The leases accepted by each lead node are classified into 5 classes based on their aggr_score. So, 5 classes are present in both the clouds. The lease to be serviced is selected based on the proposed RPCA. Fig 10 shows the comparison of the proposed RPCA along with the hybrid load balancing (centralized load balancing with intra-cloud and distributed load balancing with inter-cloud) and without load balancing for the system oriented performance parameter makespan. The total execution time required to complete all the users jobs in cloud is defined as makespan. The result shows that the better reduction in makespan by the proposed RPC algorithm with load balancing.

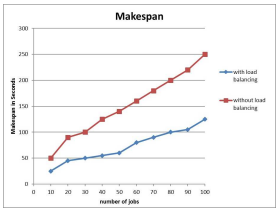

Fig. 9: Comparison of RPCA with load balancing and without load balancing for makespan

Because of the proposed monitoring system which is the time and the event based one along with the precedence of the users' jobs, and another proposed Reduced Penalty Class Algorithm with SLA for the allocation of resources on behalf of users' requests, job_success_rate will be improved compared to the other existing conventional monitoring and resource allocation methods, which in turn equivalently shows the improvement in the User Satisfaction Level (USL). The job_success_rate computation is given in (6) as follows.

$$Job\_success\_rate = \frac{N_{Success}}{T} \tag{6}$$

Here, N success represents the number of customers' jobs completed successfully and T represents the total number of customers' jobs submitted to the cloud.

6 Conclusion and Future Work

The proposed system is implemented using Open Nebula cloud tool and Xen hypervisor. The proposed monitoring algorithm is based on time and event along with the precedence of the resources. The result of the proposed PMB monitoring algorithm shows that the improvement in maintaining the accurate status of the resources for lease allotment with improved update and missed update rate performance metrics. The proposed Reduced Penalty Class Algorithm(RPCA) with the SLA formation along with negotiation shows the improvement in reduction of penalty of cloud provider and cloud user in case of violation of the agreed upon SLA parameters. So, user satisfaction level and the profit of cloud provider got improved by the proposed RPC algorithm. Further, the proposed hybrid load balancing algorithm along with RPC improves the makespan of cloud. The intermediary agents in each of the cloud helps in negotiation and SLA formation for cloud service provisioning between cloud provider and user. In addition, it performs monitoring the resource status, taking care of the reduction in the violation of SLA parameters and penalty computation and balancing the load among the cloud site with tolerable overhead.

The future work will be dealt with the time spent on negotiation, SLA formation, monitoring and load balancing should be reduced. The network utilization can be taken into account when the lease request is migrated between clouds.

Acknowledgement

The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] Belabbas Yagoubi, Medebber.M," A load balancing model for grid environment," Computer and information sciences, iscis 2007. 22nd international symposium, vol., no., pp.1-7, 7-9 Nov. 2007.

[2] Hien Nguyen Van, Frederic Dang Tran, Jean-Marc Menaud, SLA-aware Virtual Resource Management for Cloud Infrastructures, Computer and Information Technology, 2009. CIT '09. Ninth IEEE International Conference , vol.1, no., pp.357-362, 11-14 Oct. 2009.

[3] Hua-Feng Deng, Yun-Sheng Liu, Ying-Yuan Xiao2 "A Novel Algorithm for Load Balancing in Distributed Systems," Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference, vol.3, no., pp.15-19, July 30 2007-Aug. 1 2007.

[4] Iman Barazandeh,S.S. Mortazavi,A.M. Rahmani, "Two new biasing load balancing algorithms in distributed systems," Internet, 2009. AH-ICI 2009. First Asian Himalayas International Conference, vol., no., pp.1-5, 3-5 Nov. 2009.

[5] Ivona Brandic, Vincent C. Emeakaroha, Sandor Acs, Attila Kertesz, Gabor Kecskemeti, LAYSI: A Layered Approach for SLA-Violation Propagation in Self-manageable Cloud Infrastructures, Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual , vol., no., pp.365-370, 19-23 July 2010.

[6] Jacob Honore Broberg, Kartheepan Balachandran, Seren Hede, Jesper Pedersen, M. Tahir Riaz, Jens Myrup Pedersen, "Load balancing in grid networks," Advanced Communication Technology (ICACT), 2010 The 12th International Conference , vol.2, no., pp.1041-1046, 7-10 Feb. 2010.

[7] Janhavi B ,Sunil Surve, Sapna Prabhu, "Comparison of Load Balancing Algorithms in a Grid," Data Storage and Data Engineering (DSDE), 2010 International Conference on , vol., no., pp.20-23, 9-10 Feb. 2010

[8] Jian Wang, Yan Zhao, Shuo Jiang, Jiajin Le, "Providing privacy preserving in Cloud computing," Human System Interactions (HSI), 2010 3rd Conference , vol., no., pp.472-475, 13-15 May 2010

[9] K Hima Prasad, Tanveer A Faruquie, L Venkata Subramaniam, Mukesh Mohania, Girish Venkatachaliah, Resource Allocation and SLA Determination for Large Data Processing Services Over Cloud, Services Computing (SCC), 2010 IEEE International Conference on , vol., no., pp.522-529, 5-10 July 2010.

[10] Kai Hwang, Sameer Kulkarni, Yue Hu, "New Network Security Based on Cloud Computing," Education Technology and Computer Science (ETCS), 2010 Second International Workshop , vol.3, no., pp.604-609, 6-7 March 2010

[11] Keerthana Boloor, Rada Chirkova, Timo Salo and Yannis Viniotis, Heuristic-based request scheduling subject to a percentile response time SLA in a distributed cloud, GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference , vol., no., pp.1-6, 6-10 Dec. 2010.

[12] S. S. Manvi, M. N. Birje, Wireless Information Systems Research Lab "Device Resource Monitoring System in Wireless Grids," Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT '09. International Conference, vol., no., pp.260-264, 28-29 Dec. 2009.

[13] Martin Randles, David Lamb, A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing," Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference , vol., no., pp.551-556, 20-23 April 2010

[14] Min Li, Yisheng Zhang, "HPC Cluster Monitoring System Architecture Design and Implement," Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference , vol.2, no., pp.325-327, 10-11 Oct. 2009

[15] Mohammed Alhamad, Tharam Dillon, Elizabeth Chang, Conceptual SLA Framework for Cloud Computing, Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on , vol., no., pp.606-610, 13-16 April 2010.

[16] Mohd Farhan Md Fudzee and Jemal H. Abawajy, QoS-based adaptation service selection broker, Future Generation Computer Systems Volume 27, Issue 3, March 2011, Pages 256-264 .

[17] Roger Halbheer, Doug Cavit, Cloud Computing Security considerations, Security Strategist Lead, Trustworthy Computing, USA, January 2010.

[18] Ruchir Shah, Bhardwaj Veeravalli and Manoj Misra, "On the Design of Adaptive and Decentralized Load Balancing Algorithms with Load Estimation for Computational Grid Environments," Parallel and Distributed Systems, IEEE Transactions, vol.18, no.12, pp.1675-1686, Dec. 2007

[19] Sebastian Hudert, Heiko Ludwig, GuidoWirtz, Negotiating SLAs-An Approach for a Generic Negotiation Framework for WS-Agreement, Journal of Grid Computing (2009) Volume: 7, Issue: 2, Pages: 225-246.

[20] Shirlei Aparecida de Chaves, Carlos Becker Westphall and Flavio Lamin, SLA Perspective in Security Management for Cloud Computing, Networking and Services (ICNS), 2010 Sixth International Conference on , vol., no., pp.212-217, 7-13 March 2010.

[21] Stefano Ferretti, Vittorio Ghini, Fabio Panzieri, Michele Pellegrini, Elisa Turrini, QoSaware Clouds, Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on , vol., no., pp.321-328, 5-10 July 2010

[22] Ms.G.Sumathi, R.Abirami, "Resource Monitoring in Grid," Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference, vol., no., pp.361-366, 16-18 July 2008

[23] P.K.Suri, Manpreet Singh, "An efficient decentralized Load Balancing Algorithm for grid," Advance Computing Conference (IACC), 2010 IEEE 2nd International , vol., no., pp.10-13, 19-20 Feb. 2010

[24] Takahiro Hirofuchi, Hirotaka Ogawa, Hidemoto Nakada,Satoshi Itoh, Satoshi Sekiguchi, "A Live Storage Migration Mechanism over WAN for Relocatable Virtual Machine Services on Clouds," Cluster Computing and the Grid, 2009. CCGRID '09. 9th IEEE/ACM International Symposium, vol., no., pp.460-465, 18-21 May 2009

[25] Trieu C. Chieu, Ajay Mohindra, Alexei A. Karve , Alla Segal "Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment," e-Business Engineering, 2009. ICEBE '09. IEEE International Conference, vol., no., pp.281-286, 21-23 Oct. 2009

[26] Thomas Rings, Geoff Caryer, Julian Gallop,Jens Grabowski, Tatiana Kovacikova, Stephan Schulz, Ian Stokes-Rees, Grid and Cloud Computing: Opportunities for Integration with the Next Generation Network, J Grid Computing, vol., no.,7- 3, pp. 375-393 2009.

[27] Vincent C. Emeakaroha, Ivona Brandic, Michael Maurer, Schahram Dustdar, "Low level Metrics to High level SLAs - LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments," High Performance Computing and Simulation (HPCS), 2010 International Conference, vol., no., pp.48-54, June 28 2010-July 2 2010

[28] Weili Huang, Jian Yang, "New Network Security Based on Cloud Computing," Education Technology and Computer Science (ETCS), 2010 Second International Workshop , vol.3, no., pp.604-609, 6-7 March 2010.

[29] Wu-Chun Chung, Ruay-Shiung Chang, Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan, ROC- A new mechanism for resource monitoring in Grid computing-Future Generation Computer Systems 25 (2009) 17.

[30] Xioojiang Duo, "Toward efficient distributed network monitoring," Performance, Computing, and Communications, 2004 IEEE International Conference , vol., no., pp. 87- 94, 2004
[31] Yi Zhao, Wenlong Huang, "Adaptive Distributed Load Balancing Algorithm Based on Live Migration of Virtual Machines in Cloud," ncm, pp.170-175, 2009 Fifth International Joint Conference on INC, IMS and IDC, 2009.
[32] Jorge E. Pezoa, Sagar Dhakal, Majeed M. Hayat. Maximizing Service Reliability in DistributedComputing Systems with Random Node Failures: Theory and Implementation IEEE Transactions on Parallel and Distributed Systems 21(10); 1531-1544; 2010.
[33] M. Elena Renda, Giovanni Resta, and Paolo Santi. Load Balancing Hashing in Geographic Hash Tables. IEEE Transactions on Parallel and Distributed Systems 23(8); 1508-1519; 2012.
[34] Hung-Chang Hsiao, Hsueh-Yi Chung, Haiying Shen, Yu-Chang Chao. Load Rebalancing for Distributed File Systems in Clouds. IEEE Transactions on Parallel and Distributed Systems 24(5); 951- 962; 2013.
[35] Dinil Mon Divakaran, Tho Ngoc Le, Mohan Gurusamy. An Online Integrated Resource Allocator for Guaranteed Performance in Data Centers. . IEEE Transactions on Parallel and Distributed Systems 25(6); 1382- 1392; 2014.
[36] Dario Bruneo, Salvatore Distefano, Francesco Longo, Marco Scarpa. Stochastic Evaluation of QoS in Service-Based Systems. IEEE Transactions on Parallel and Distributed Systems 24(10); 2090- 2099; 2013.
[37] Hossein Morshedlou, Mohammad Reza Meybodi. Decreasing Impact of SLA Violations: A Proactive Resource Allocation Approach for Cloud Computing Environments. IEEE Transactions on Cloud Computing 2(2); 156-167; 2014.
[38] Mario Macias, Jordi Guitart. SLA negotiation and enforcement policies for revenue maximization and client classification in cloud providers. Future Generation Computer Systems 41;19-31; 2014.
[39] Saurabh Kumar Garg , Adel Nadjaran Toosi , Srinivasa K. Gopalaiyengar , Rajkumar Buyya. SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter. Journal of Network and Computer Applications 45; 108-120; 2014.
[40] Alexander Stanik, Marc Koernera, Leonidas Lymberopoulos. SLA-driven Federated Cloud Networking: Quality of Service for Cloud-based Software Defined Networks. Procedia Computer Science 34; 655- 660; 2014.
[41] Kwang mong Sim, Agent-Based Cloud Computing, IEEE Transactions on Services Computing, 5;99; 2011.
[42] Muthurajkumar, S, Vijayalakshmi, M, Kannan, A, Secured Temporal Log Management Techniques for Cloud, Procedia Computer Science,46, 589 595, 2015.
[43] Muthurajkumar, S., Vijayalakshmi, M., Kannan, A., Intelligent Temporal Role Based Access Control for Data Storage in Cloud Database. 2014 Sixth International Conference on Advanced Computing(lCoAC), IEEE Digital Library. pp. 184-188, 2015.
[44] P.Seenuvasan, R.Geethramani, P.Varalakshmi, Renuka Ramachadran, Energy-Efficient Server-Consolidation Based Resource Allocation in Cloud, First International Conference on Mathematical And it's Application. Vol. 3-1888 1897,2014.

Seenuvasan P is working as Assistant Proffessor in the Department of Information Technology, University College of Engineering, Villupuram, A constituent college of Anna University, Chennai. His areas of interest includes Cloud Computing, Data Base Management System and Computer Networks.

Kannan A is working as Professor in the Department of Information Science and Technology, College of Engineering Guindy, Anna University, Chennai. He has published more than 300 papers in various reputed journals and conference proceedings. His areas of interest includes Data Base Management System, Artificial Intelligence, Computer Networks and Security, Wireless Sensor Networks, Big Data Analytics and Cloud Computing.

Varalakshmi P is working as Associate Professor in the Department of Computer Technology, Madras Institute of Technology, Anna University, Chennai. She has published more than 100 papers in various reputed journals and conference proceedings. Her areas of interest includes Cloud computing, Networks, Security, Wireless Sensor Networks, Mobile Computing,Compiler,Theory of Computation and IOT.