Secure VM Migration to Improve Load Balancing in Cloud Environment

[1]S.Jyothsna, [2] Dr.K.Radhika
[1]Asst.Professor, [2]Professor
[1] IT Department,
[1]CVR College of Engineering, Hyderabad, India

Abstract : Cloud datacenters provide powerful computing power and required resources on demand but the average utilization of cloud servers is comparatively low. Some servers may be overloaded and others may be underutilized. Efficient Load balancing is required to improve the resource utilization and minimize the downtime. The virtual machine migration is one of the key aspect in cloud to balance the load among cloud servers. Securing the virtual machine (VM) in migration , minimizing migration time remain open issues. Virtual machine security issues need to be resolved when migrating a virtual machine in cloud. XACML can be used for defining role based access control policies for a VM to provide accessibility only to the authorized resources or users and proposed an XACML based algorithm for load balancing to improve the performance and security of the cloud services.

Index Terms **- Load Balancing, Cloud Servers, VM Migration, Security issues, XACML**.

I. INTRODUCTION

Cloud computing is Internet based computing for providing quality services to the users from any geographical location. Scalability is one of the challenging QOS parameter in cloud as the cloud is providing various hardware and software services through public or private clouds. Virtualization has improved the resource provisioning in cloud as the user can access the services from virtual machine and the assignment of tasks to virtual machine is the responsibility of the scheduling algorithm. The efficient scheduling must be done for task assignment by monitoring the load on the servers after every new assigned task and accordingly the load must be distributed among the number of servers otherwise the servers may lead to the issues of over loaded node (OLN) or under loaded node(ULN).To balance the load, The load must be transferred or migrated from one server to another which leads to the security issue. Dynamic consolidation of virtual machines (VMs) using live migration and switching idle nodes to the sleep mode allow cloud providers to optimize resource usage and reduce energy consumption.

The Assignment of tasks to VMs must consider the load of that physical machine and task assignment decision must be taken based on that load. Load balancing algorithms plays an important role in balancing the load among the cloud servers. Load balancer is the part of scheduling. The load of each cloud server must be monitored regularly and VMs. need to be migrated to balance the load among cloud servers.

Load Balancing can be static or dynamic. Static load balancing algorithms need prior knowledge of number of tasks need to be serviced and availability of number of servers. In Cloud Computing tasks need to be serviced dynamically so the static load balancing algorithms may not give efficient results. The dynamic load balancing algorithm is required which can schedule the tasks dynamically and balance the load accordingly among the cloud servers.

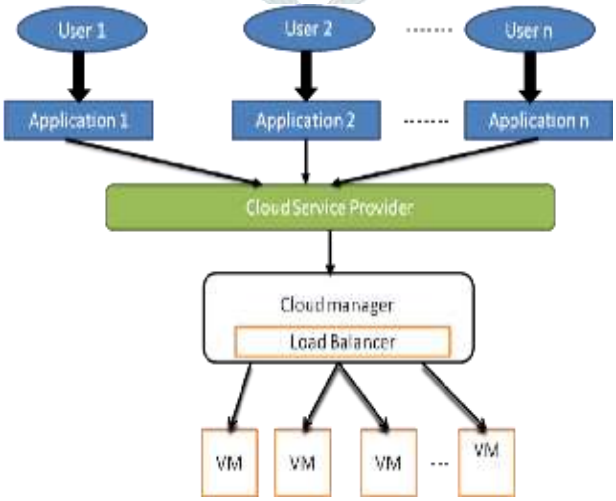

Fig.1 Load balancing in cloud

Challenges in Load Balancing

1. Overhead Associated: Overhead due to movement of tasks, inter process communication, overhead should be reduced so that load balancing algorithm works well.

2. Throughput: It is the amount of work done in fixed interval of time. Most of the algorithms tries to improve the throughput.

3. Resource Utilization: Resource utilization should be maximum for an efficient load balancing algorithm.

4. Scalability: The quality of service provisioning should not be effected as the number of users/jobs increases. The more number of nodes can be added flexibly without affecting the service.

5. Response time: Minimum response time is expected by the users as per service level agreement (SLA).

Limited resources on same physical machine running multiple VMs causes resource conflict due to which physical machine may fail to serve continuously. Hence, to avoid failures, VM migration is the solution to have continued and uninterrupted service. Different Migration techniques were proposed by various researchers to control the overloaded or under loaded issues in cloud computing. VM migration is to migrate memory and control of VM from one physical server to another without any service interruption. Downtime and total migration time should be considered when the migrated VM is in working condition.

VM migration can be active or passive.

Passive/Cold VM Migration: Migration of a powered-off virtual machine from current host/data store or both to a new host or data store.

Active/Live VM Migration: Live migration transfers the VM from one physical server to another without disconnecting/powered-off with the client so that downtime can be reduced and performance of cloud services can be improved using continuous availability of services which helps System administration in fault tolerance, online system maintenance, workload balancing, consolidation of VMs etc. However, vulnerabilities associated with live migration pose many security threats.

Live migration is an essential feature of virtualization, defined as a process of dynamically transferring running VMs from one physical server to another with little or zero downtime and without interrupting services running in VM. Downtime is the total time for which VM stops running. Most of the Load balancing algorithms were tried to minimize the make span and improve the throughput but VM migration and security issues are still need to be improved.

Benefits of Live Migration

1. Service Providers can be changed flexibly, can take advantage of low-price.

2. Offering service continuity in case of ceasing operation or natural disasters.

3. Cloud users can be connected to the nearest datacenter, regardless of the provider.

4. Processing sensitive data on a private trusted cloud while processing less sensitive data on a public cloud.

5. Borrowing resources from different providers in case of over-utilization or limited resources

All VM properties and attributes remain unchanged, in live migration including internal and external IP addresses, instance metadata, block storage data and volumes, OS and application state, network settings, network connections, and so on. In live migration process there are several authentication issue as well as active and passive attacks which exploits live migration process. The Main cause of this is lack of secure live migration protocol. Hence secure live migration protocol must be required for live migration having essential features like protected transmission channel, integrity of migration data and entity authentication. XACML role based access control policy is proposed for virtual machine while migrating from one server to another as XACML allow natural integration with the Cloud and Web Services security services infrastructure.

The rest of paper is organized as follows. Section II provides related work of load balancing algorithms, section III provides the related work pertaining to VM migration and security issues of live migration of VM, Section IV problem formulation, section V provides the proposed system design and section VI is conclusion and future work.

II. Related Work

Many load balancing algorithms were proposed in cloud computing and compared QOS parameters like response time and resource utilization. The Load balancing algorithms can be static or dynamic.

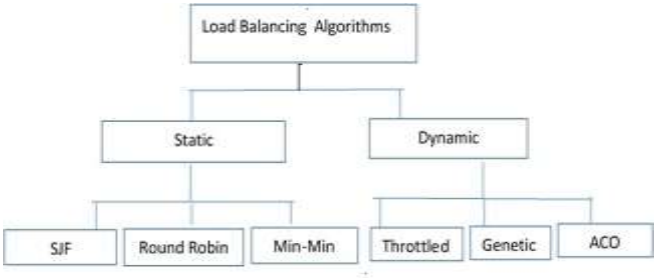

Fig.2 Load Balancing Algorithms

SJF: The main idea of the Modified Shortest Job First [1] is to sort the tasks in an ascending order based on the task length and calculate the average of all the tasks length. Then for all tasks the algorithm checks each task length, if it is less than the average tasks length and number of tasks in VM1 less than the number of tasks in VM2, then the task will be sent to VM1, else to VM2. It does not only focus on the completion time of the task but also takes into account the total completion time of all tasks.

Round Robin: Nithin Das K.C, Melvin S George and Jaya P [3] incorporated Weighted Round Robin algorithm in Honeybee algorithm in order to achieve minimum data center processing time and response time. For the tasks with priority, Honeybee Inspired algorithm is used by assigning weights to each virtual machine and the virtual machine is selected according to the resource requirement of the tasks. Tasks with no priority are executed using Weighted Round Robin algorithm. The proposed algorithm shows better results when compared with Weighted Round Robin and Honey bee Inspired load balancing algorithm.

Min-Min: Shyam Singh Rajput and Virendra Singh Kushwah [4] proposed Improved Load Balanced Min-Min (ILBMM) algorithm using genetic algorithm (GA) in order to minimize the make span and increase the utilization of resource. the execution time of task on the virtual machine was calculated then found the minimum or maximum time of task on virtual machine (VM), for better execution, genetic based approach is applied on million instructions (MI) of task and million instruction per second (MIPS) of VM. The results shows that the algorithm has minimized the make span compared to the existing LBMM algorithm.

Throttled: Soumi Ghosh Chandan Banerjee and Netaji Subhash proposed a modified priority based throttled algorithm [2] where The queue is used to hold the requests which are switched from execution state to wait state. The switching of task is done depending on the priority of the requested task. If the priority of the new request is greater than the priority of the executing request, then the executing request is switched by the new request and placed in Queue. If higher priority requests are served continuously, then the lower priority would never get a chance to be executed; i.e. it would suffer from starvation. To overcome this problem, priority of each waiting request is increased. If any of the Virtual Machine gets free, then it would check for new service request and the waiting request, then selects the job having highest priority and allocate VM to it. The Modified Priority Throttled Algorithm focuses mainly on the assignment of incoming jobs to the available virtual machines and distributed load uniformly among VMs depending on priority of the task, Results shows that resource utilization and response time has improved.

Genetic Algorithm: The Genetic Algorithm (GA) is an optimization algorithm which uses the method of computerized search based on natural selection and genetics.

ACO and Genetic Approach: Ashish Gupta and Ritu Garg[5] proposed an algorithm which is based on behavior of ant colonies for searching food and connecting with one another through pheromone trails that are left behind by ants on the paths they travel. In this paper, The ACO approach is used for scheduling independent tasks on the set of available resources with the aim to optimize makespan along with load balancing. Make span is the maximum completion time of all the tasks allotted to different machines. Ants construct solutions to scheduling problem during an iteration by moving from one VM to another until the tour is completed (i.e., until all tasks have been placed). Each mapped task onto a specific resource find pheromone and heuristic information and resource is picked up based on pheromone trails. For successive ants, initial pheromone values will be performed on updated pheromone values for previous ants. Local solutions for set of ants with makespan and load balancing level is obtained. Non dominated sort genetic algorithm is applied on multi-objective(minimize make span and load balancing) solutions to find the best local solutions. After every iteration, global solution is updated with pheromone matrix and best local solutions is generated. Procedure repeats and optimal solutions are found in the global solution after maximum number of iterations.

Elastic and flexible deadline constraint load balancing algorithm: Mohit Kumar, Kalka Dubey and S.C. Sharma,[6] proposed a cloud architecture that is capable of handling maximum user requests before meet the deadline and provides an elasticity mechanism with the help of threshold based trigger strategy. It calculate the number of tasks unable to meet deadline in each interval after that average of rejected task in last z interval is calculated and apply the user defined threshold conditions as per SLA.

If value of rejected tasks is more than or equal to the value of 30% of total tasks then 20% new VM will be added. If value of rejected list is more than 10% of total tasks then 10% new VM will be added. If rejected list is less than or equal to10% then there is no need to add the new VM. A VM is considered overloaded mode if they utilize their capacity more than or equal to 90% and under loaded if utilize their capacity less than 20%. Sorts VM in decreasing or increasing order based on overloaded or under loaded condition and transfer the task from overloaded node to under loaded node. If average of under loaded virtual machine is greater than the 30 % of all available virtual machine then decrease the virtual machine 20 % for next interval. If it is more than 10% then decrease the virtual machine 10% for next interval. Computational results shows that this algorithm reduce the make span time compared to FCFS, SJF and Min-Min algorithm.

Ant colony optimization (ACO) and Genetic Algorithm (GA) suffers from poor convergence speed. Whereas Particle swarm optimization has good speed due to velocity but limited to the initial set of particle problem. honeybee colony is improved further using the variance among the schedules[9]. Also, the multi objective fitness function is also designed to enhance the results further. The proposed technique initially schedules the jobs on high end servers and then try to balance the load between these high end servers.

III. VIRTUAL MACHINE MIGRATION

After scheduling the tasks, load balancing at run time will come in action [9]. Load balancing will migrate some jobs of heavily loaded high end servers (HESs) to under loaded HESs. Highest priority is given to jobs with minimum burst time. Scout bees are responsible for evaluating the over and under loaded HESs.

The Following Parameters must be considered while migrating a VM from one host to another host.

1. ***Total Migration Time:*** It indicates the time from the sender host enters migration process to time the destination host finishes it and starts working normally.

2. ***Network Traffic Reduction Percentage:*** The amount of data transfer saved because of the memory page compression and deduplication of it during live migration.

3. ***Downtime:*** The time duration up to which the VM is actually suspended to transfer CPU state as well as transfer WWS to the destination host.

4. ***Application Degradation***: It is the extent up to which live migration has slow down the application performance of migrating VM authorities.

Migration can take place at different levels

A. Process Migration: In process migration, process moves from one physical server to another physical server. In 1980's more research was done in process migration. However, due to residual dependency process migration didn't get popularity.

B. OS Migration: OS migration is another approach which handles all limitation of process migration and does the virtual machine migration efficiently. OS migration overcomes the residual dependency problem and administrator need not worry about it. Administrator can migrate OS and its associated process as single unit.

C. VM Migration: Migration requires that each memory state is stored consistently at application level state and kernel internal state on the target machine. This complete process degrades the cloud performance for a specific amount of time and may disappoint an active user. Even though this technique helps much in improving load balancing in cloud environment its downtime is the main negative effect.

Live migration keeps VM instances running during:

- Regular infrastructure maintenance and upgrades.
- Network and power grid maintenance in the data centers.
- Failed hardware such as memory, CPU, network interface cards, disks, power, and so on.. if a hardware fails completely or otherwise prevents live migration, the VM crashes and restarts automatically.
- Host OS and BIOS upgrades.
- Security-related updates, with the need to respond quickly.
- System configuration changes, including changing the size of the host root partition, for storage of the host image and packages.

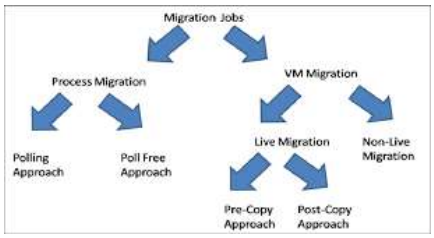

Fig.3 Types of Migration

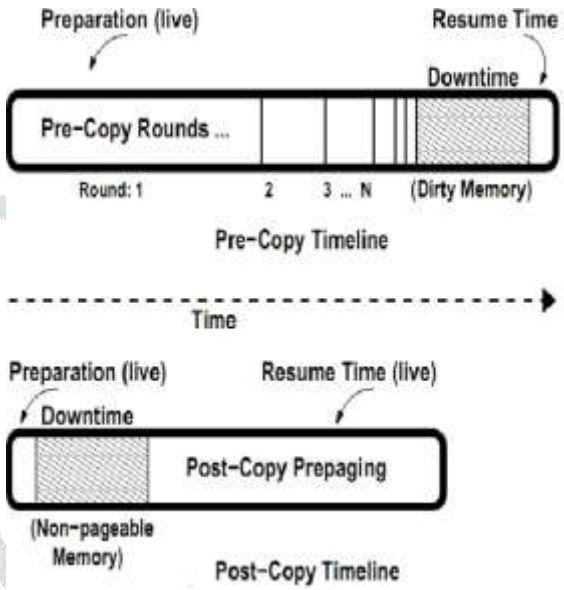

Fig.4 Migration Approaches

The migration approach can be

Pre Copy Approach: In this type of live migration, memory transfer phase starts with copying memory pages while the VM is running at the sender host. If in the i^{th} round a memory page is dirtied it is sent again to the destination host in i+1th round of page transfer. The Load of the destinatin host is calculated in each round. This iterative process continuous until the destination host overloaded.

***Post Copy Approach*:** Post copy live migration first copies sender processor state at source host and resumes the VM CPU state at the destination host. And after resuming VM at destination host, post copy live migration technique starts memory page transfer.

Security Issues of Live Migration

Live migration might be susceptible to range of attacks from Denial-of-Service (DoS) attacks to Man-In-The-Middle (MITM) attacks. During the migration, data can be tampered or sniffed easily as it is not encrypted. Thus compromising confidentiality and integrity of migrating data. These security threats in live VM migration discourages many sectors, such as financial, medical, and government from taking advantage of VM live migration. Hence, security is the critical challenge that needs examination to provide secure live VM migration.

A secure live migration of VM requires trusted source and destination platforms, authentication and authorization mechanism, confidentiality and integrity of migrated data, Mechanism to detect and notify suspicious activities. Open source hypervisors like VMware‘s (VMotion), Xen, KVM.Oracle‘s Virtual box etc supports live migration. Most of the hypervisors are performing live migration manually with a little or no consideration towards its security.

There are several security loop holes in live migration done by using KVM, Xen and VMware hypervisor. For Ex: VMware may expose the sensitive information during migration and the Xen can take advantage of vulnerability in migration module and hence can take complete control of VMM or host VM. Migration protocol used is not secure and does not encrypt migration data of VM. Hence, there is no confidentiality of migrated data, other vulnerabilities like untrusted platform, authentication, authorization

and bugs in hypervisor code etc..

Current techniques face many challenges while migrating memory and data intensive applications like network faults, changing the network connection type, and changing the virtual hardware compatibility. Consumption of network bandwidth and cloud resources are some of the attacks at cloud infrastructure level. Restrictions can be enabled to prevent users from removing virtual devices, changing the memory allocation, modifying fixed values of utilization thresholds. These are unsuitable for an environment with dynamic and unpredictable workloads, in which different types of applications can share a physical resource. The system should be able to automatically adjust its behavior depending on the workload patterns exhibited by the applications

a. The attacker may steal the bandwidth by taking control of the source virtual machine and migrating it to the destination virtual machine.

b. The attacker may falsely advertise its resource and attract others to migrate its resources towards itself.

c. Active manipulation: Attacker may modify the data which is travelling from the source to the destination.

d. Passive snooping: Attacker just accesses the data of migration using any sniffing tool that may lead to leakage of some confidential information.

Existing Security Controls

VM is encrypted with AES algorithm before migrated [10].VM can be accessed by the user if the authentication is provided so that security features like confidentiality, integrity and authentication is provided. VM is encrypted using AES algorithm using 256 bit key in Xor-Encrypt-Xor(XEX) bases tweaked codebook mode with cipher stealing (XTS) mode. Tweak value is a 128bit data used to represent the logical position, which may be encrypted or decrypted with XTS-AES. When the user tries to login, authorization is required. It is required to provide the password to decrypt the VM and launch VM to work upon it. It's difficult to gather the password by an unauthorized user. Only legitimate users are allowed to access and decrypt the VM image even after migration. The drawback of this approach is the VM has to be power down in order to encrypt it so it can't be live migration. The downtime may be increased.

Live cloud migration (LivCloud) [11] used the User Datagram Protocol based data transfer and KVM to enable nested virtualization on the cloud IaaS as well as securing the migration channel. To maintain encryption and authentication, LivCloud uses VPN, VPN is Amazon Virtual Network that helps building user-defined private network subnets inside the cloud in order to facilitate controlling IP address changes. This is able to provide encryption using Advanced Encryption Standard (AES) and authentication using Hash based Message Authentication Code (HMAC-SHA1. Optimization is needed to tackle the migration negative impact when live migrating between different hypervisors KVM and VMware and when VM disks are hosted on an NFS server.

.

IV. PROBLEM FORMULATION

The VMs experience dynamic workloads in cloud, which means that the CPU usage by a VM arbitrarily varies over time. The host is oversubscribed, i.e., if all the VMs request their maximum allowed CPU performance, the total CPU demand will exceed the capacity of the CPU. When the demand of the CPU exceeds the available capacity, a violation of the SLAs established between the resource provider and customers occurs. To balance the load among cloud servers the migration of VM may take place.

The migration of VM can be categorized as follows based on the number of VMs to migrate.

Single VM migration problem- a single VM can be migrated out from the host. This migration leads to a decrease of the demand for the CPU performance and makes it lower than the CPU capacity.

Dynamic VM consolidation Problem- This is a more complex problem considering multiple hosts and multiple VMs. There are n homogeneous hosts are defined and the capacity of each host is A_h .VMs experience variable workloads, The maximum capacity that can be allocated to VM is A_v. Therefore, the maximum number of VMs allocated to a host when they demand their maximum CPU capacity is $m = A_h / A_v$. The total number of VMs is n_m. VMs can be migrated between hosts using live migration with a migration time t_m. An SLA violation occurs when the total demand for the CPU performance exceeds the available CPU capacity A_h. This host can be considered as overloaded host where there is a need of migrating a set of VMs from this host to another host (destination host) until The host is not being considered as overloaded.

After deciding the VMs to migrate, one more issue is to find the underutilized host or idle host as destination host so that load can be balanced and migrating a VM using appropriate security controls.

VM Security through Access Control Policy

Appropriate access control policies must be provided to secure the live VM migration process. An unauthorized user/role may launch VM initiate, migration operation. Unauthorized activities can be prevented by using access control list (ACL's). The Extended Access Control Markup Language (XACML) is used to apply access control policies to VM. XACML is an OASIS standard that describes both a policy language and a role based access control decision request/response language.

V. SYSTEM DESIGN and SIMULATION TOOLS

In Cloud Computing, The load balancing problem can be considered as multi objective NP-complete problem. The objectives may be efficient resource utilization, make span minimization, throughput maximization, etc. The proposed algorithm can be implemented using a cloud simulation tool cloudsim3.0 which facilitates to set up a cloud environment by implementing the Datacenter ,DatacenterBroker, Host, Vm and Cloudlet classes etc. different test cases will be evaluated and compare the results with existing algorithms. The algorithm starts with calculating load on each physical machine using following.

Load of a physical machine= Σ (capacity in MIPS) / Σ(assigned task length in MI)

The overloaded node (OLN) can be detected using algorithm1 selects the virtual machine"VM" from OLN and migrate it to under loaded node(ULN) based on either of the following policies

1. Migration time: The migration time can be calculated for each virtual machine then selects the VM which needs less amount of time to migrate so that the downtime can be improved.

2. Random Selection: The VM can be selected randomly so that each VM gets an equal opportunity and easy to apply this policy but this may not work efficiently as it is not considering any parameters for selection.

3. Memory Consumption: The virtual machine which occupies more memory can be selected to migrate so that the memory can be consumed efficiently by avoiding the memory overhead.

When migrating virtual machines from source host machine, access control policies can be used to protect virtual machine both at rest and in transit, just like any other data we store and transmit. secure policy is applied to the selected VM while migrating using algorithm2.

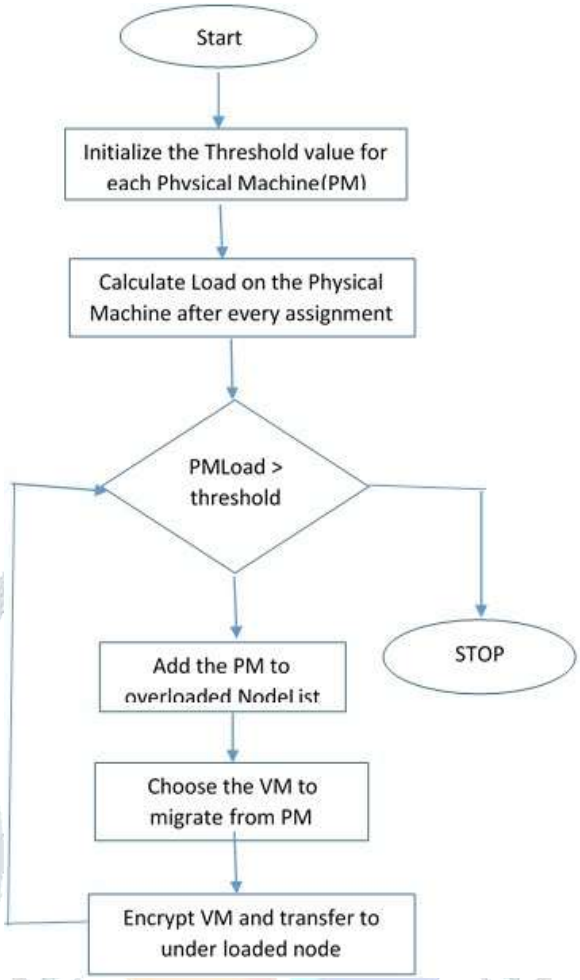

Fig.5 Flowchart

PROPOSED ALGORITHM

Algorithm 1 : Load Balancing using secure Live VM Migration
Input

VM List: Number of virtual machines
PMList: Number of Physical machines
Threshold(t): The Threshold value of each Physical Machine(PM) based on the total CPU consumption.

Output
Balanced Servers or Physical nodes in cloud

Procedure

repeat
1. Identify the overloaded host.
2. Select particular VMs to migrate from this overloaded host.
3. Apply the access control policies for VM using XACML(VM-XACML)
4. Migrate selected VM to another Host

until (host is considered as not overloaded)

Algorithm 2: Access control policies for VM using XACML(VM-XACML)

XACML suits for expressing access control policies to complex distributed resources with different user access rules that also may require domain based hierarchical user roles and permissions management. The XACML Role based access control policy provides extended functionality for managing user/subject roles and permissions by defining separate Permission <PolicySet>, Role <PolicySet>, Role Assignment <Policy>, and HasPrivilegeOfRole <Policy>.

The typical setup is that someone wants to take some action on a virtual machine, so the elements of request are user attributes, action and resource to be accessed The request/response language form a query to ask whether or not a given action should be allowed on the VM and interpret the result. The response always includes answer about whether the request should be allowed using one of four values: Permit, Deny, Indeterminate or Not Applicable. A sample of request and response can be shown as follows.

```
<Request>
 <Attributes>
  <Attribute AttributeId=”User1">
   <AttributeValue  DataType="http://www.w3.org/2001/XMLSchema#string">Jyothsna
   </AttributeValue>
 </Attribute>
</Attributes>
</Request>

<Response>
  <Result>
   <Decision>Permit</Decision>
    <Status>
     <StatusCode Value="urn:oasis:names:tc:xacml:1.0:status:ok"/>
    </Status>
 </Result>
</Response>
```

VI. CONCLUSION AND FUTURE WORK

Load balancing importance in cloud and its challenges, various load balancing algorithms, VM migration techniques and security issues in VM migration were reviewed in this paper and proposed a secure algorithm for VM migration and load balancing using XACML. The proposed algorithm tries to control the mentioned security issues while balancing the load using secure VM migration. The cloud servers are monitored regularly and tries to balance the load among them by the process of VM migration. The access policies must be defined to VM to avoid security issues. Planning to implement and test the proposed algorithm with various test cases in future.

REFERENCES

[1] Mokhtar A. Alworafi, Atyaf Dhari, Asma A Al-Hashmi , Suresha and A. Basit Darem “An Improved SJF Scheduling Algorithm in Cloud Computing Environment”,IEEE,2016 International Conference on Electrical, Electronics,Communication, Computer and optimization Techniques(ICEECCOT) ,9-10 Dec. 2016, **DOI:**10.1109/ICEECCOT.2016.7955216

[2] Nithin Das K.C, Melvin S George and Jaya P,“Incorporating Weighted Round Robin in Honeybee Algorithm for Enhanced Load Balancing in Cloud Environment”,IEEE,International Conference on Communication and Signal Processing, April 6-8, 2017, India.

[3] Shyam Singh Rajput and Virendra Singh Kushwah, “A Genetic based Improved Load Balanced Min-Min Task Scheduling Algorithm for Load Balancing in Cloud Computing”,2016 8th International Conference on Computational Intelligence and Communication Networks, 23-25 Dec. 2016.

[4] SoumiGhosh·ChandanBanerjee,Netaji Subhash Engineering College, Kolkata “Priority Based Modified Throttled Algorithm in Cloud Computing”,2016 International Conference on Inventive ComputationTechnologies(ICICT)**DOI:**10.1109/INVENTIVE.2016.783017510.1109/INVENTIVE.2016.7830175

[5] Ashish Gupta and Ritu Garg,“Load Balancing Based Task Scheduling with ACO in Cloud Computing”,IEEE,2017 International conference on Computer andApplications(ICCA),DOI:10.1109/COMAPP.2017.8079781

[6] Mohit Kumar,Kalka Dubey and S S.C. Sharma,“Elastic and flexible deadline constraint load balancing algorithm for cloudcomputing”,ELSEVIER,6thInternational Conference on Smart Computing and Communications ICSCC 2017 ,7-8 December 2017,Procedia Computer Science 125 (2018) 717–724

[7] EssaiesMeriam and Nabil Tabbane," Multiple QoS Priority Based Scheduling in CloudComputing",IEEE,2016International Symposium on Signal, Image, Video and Communications (ISIVC), 21-23 Nov. 2016, **DOI:** 10.1109/ISIVC.2016.7894000

[8] Gibet Tani Hicham, El Amrani Chaker "Study on Different Scheduling Algorithm for Cloud Computing", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 6, No. 4, August 2016, pp. 1866~1879ISSN: 2088-8708, DOI:10.11591/ijece.v6i4.10144

[9] Neha Sethi,Dr.Surjit Singh and Dr.Gurvinder Singh,"Multiobjective Artificial Bee Colony based Job Scheduling for Cloud Computing Environment",I.J.MathematicalSciences and Computing,2018, 1, 41-55.

[10] Anitha H M and Dr.P.Jaya Rekha, "Secure Virtual Machine Migration In Virtualized Environment",IEEE, ICISC 2018

[11] Ibrahim Ejdayid A.Mansour, Kendra Cooper and Hamid Bouchachia "Effective Live Cloud Migration", 2016 IEEE 4th International Conference on Future Internet of Things and Cloud.

[12] Yuri Demchenko, Leon Gommans, Cees de Laat,"Using SAML and XACML for Complex Authorisation Scenarios in Dynamic Resource Provisioning."