Enhanced Methodology to perform Task Scheduling using Load Balancing on Cloud Environment

Lanka Vijay #1, Ch Subhash Chandra #2
#1Student of M.Tech (CSE) and Department of Computer Science Engineering,
#2 Assoc.Prof, Department of Computer Science and Engineering,
Kakinada Institute of Engineering and Technology, AP.

Abstract
Cloud Computing is a rising computing worldview. It plans to share information, computations, and administration straightforwardly finished a versatile system of nodes. Since Cloud computing stores the information and scattered assets in the open condition. In this way, the measure of information storage increments rapidly. In the cloud storage, load balancing is a key issue. It would expend a great deal of cost to keep up load data, since the framework is excessively tremendous, making it impossible to auspicious scatter load. Load balancing is one of the fundamental difficulties in cloud computing which is required to convey the dynamic workload over various nodes to guarantee that no single node is overpowered. It helps in ideal usage of assets and subsequently in improving the execution of the framework. A couple of existing scheduling algorithms can keep up load balancing and give better systems through proficient employment scheduling and asset portion methods too. So as to increase greatest benefits with improved load balancing algorithms, it is important to use assets effectively. The principle point of this paper is to talk about a existing of the current load balancing algorithms in cloud computing condition.

Keywords: Cloud Computing, Load Balancing, Task Scheduling, Quality of Service.

I. Introduction
Cloud computing enables every client to utilize the software and computing services on request whenever, in wherever and anyplace through the Internet. Cloud computing primarily manages computing, software, information access and storage services that may not require learning of the end-client's land area and framework arrangement, which is to give services [2]. Clouds show shifting requests, framework sizes, supply examples and assets (equipment, software, and system); clients have heterogeneous, dynamic, and Quality of Service (QoS) necessities; and applications have changing execution, workload, and dynamic application scaling prerequisites [3]. The primary goal of cloud computing is to give simple, adaptable approach to computing assets and IT services. Quick advancement of cloud computing shows up through numerous associations, for example, GoGrid, Google, Rack space, Microsoft, Amazon EC2 cloud computing and Apple to give cloud services to different shoppers. The cloud framework powerfully designates computing assets for the client/client because of clients' asset reservation demands and as per clients' QoS necessities [4]. The qualities of cloud computing are multi-tenure, leased services conveyance show, on-request use, outside information storage, straightforward, quick flexibility, a wide system get to, asset pooling and measured service[5, 6]. An essential piece of the cloud is the asset administration. The asset administration technique in cloud ought to adequately use the pool of asset and accomplish a high framework execution. Asset administration can be accomplished through a type of load balancing among the taking part nodes. On that point are a few measurements that will serve to quantify the proficiency of each load balancing systems (LBT). LBT in a cloud situation; consider different parameters [7, 8], for example, reaction time, throughput, adaptability, unwavering quality, QoS, asset use and adaptation to internal failure. Asset administration influences three essential criteria for framework assessment, They are the execution, usefulness and cost. Wasteful administration of assets has an immediate negative impact on execution and
cost. You can likewise in a roundabout way influence framework usefulness [9]. Matchmaking and scheduling are imperative issues performed by asset administrators in the cloud [10]. Asset distribution in a cloud situation includes two stages, matchmaking is the main stage and scheduling is the second stage. Matchmaking is characterized as, the strategy for distributing employments related with client solicitations to assets assigned from the possible asset pool. Load adjust implies disperse load of various assets to accomplish most extreme throughput, limit the reaction time and to evade the overloading at a specific node. Both matchmaking and scheduling need to fulfill clients' QoS necessities characterized in an administration level understanding (SLA).

II. Related work

In 2003, H. Xiaoshan et al [3] recommended a QoS Guided Min-Min heuristic [Batch mode heuristic algorithm] was presented in that some undertaking require higher system transfer speed to trade a lot of information among processors, though some can be happy with the lower arrange data transmission. In this calculation the coordinating of the QoS ask for and services between the undertakings and hosts in view of traditional Min-Min. Right off the bat each errand with the high QoS ask for in the Meta undertaking, the calculation finds the soonest finish time and the host that gets it, in the whole QoS Qualified host. Besides discover the undertaking with the base soonest fruition time and allocates the assignment to the host that give the most punctual finishing time to errand. In this calculation they have tended to just a single measurement QoS issue, since they worked just data transmission imperative. In 2006 F. Dong et al [4] proposed a QoS need gathering calculation which considers due date and approval rate of the errand and the makespan as principle factor of assignment scheduling in entire framework. It accomplishes better acknowledgment rate and consummation time for the submitted errand then MinMin and QoS Guided Min-Min. In 2008, C. Hsu et al [5] did two enhancement plans MOR (Makespan Optimization Rescheduling) and ROR (Resource Optimization Rescheduling). MOR concentrate on enhancing the makespan to pull off the better execution and in ROR concentrate on the redispash assignments from the machine with the base number of undertakings to other machine, which is useful to diminish the asset require. Both this system accomplishes low many-sided quality, high adequacy, great execution than QoS Guided scheduling calculation and Min-Min calculation. In 2008, M. Singh et al [6] proposed a QoS based prescient Max-Min, Min-Min switcher calculation. In this calculation, scheduling of the following employment depends on fitting choice among QoS based min-min or QoS max-min calculation. The impact on the execution time network occupations has been diminished due to non-committed assets. It ordinarily utilizes the history data about the execution occupations to foresee the execution of non-committed assets. This calculation consolidates the effectiveness of max-min alongside min-min and furthermore considers both QoS and nondedicated property of matrix assets. In 2009, S. Parsa et al [7] presented another assignment scheduling calculation called RASA which has the upside of both Min-Min and Max-Min calculation. In this initially evaluate the culmination time of the assignments on every asset and after that connected both the calculation. RASA utilize the MinMin technique to execute the little errand first at that point long undertaking and after that connected Max-Min to maintain a strategic distance from the postponements in the execution of huge assignment and bolster simultaneousness in the execution of the extensive and little errands. It accomplishes the lower Makespan with great QoS. In 2010, Mrs. S. Selvarani et al [1] presented an enhanced cost based scheduling calculation for making productive mapping of errands to accessible assets in cloud. The act of spontaneity of conventional movement based costing is proposed by new undertaking scheduling system for cloud condition where there might be no connection between the overhead application base and the way that diverse assignments cause overhead cost of assets in cloud. This scheduling calculation separates all client undertakings relying upon need of each errand into three unique records. This scheduling calculation measures both asset cost and calculation execution, it likewise Improves the calculation/correspondence proportion. In 2013, X. Wu et al [10] present an errand scheduling calculation in view of QoS-driven in cloud computing (TS-QoS). In this TS-QoS calculation register the need of the errand as per the exceptional properties of the assignments, and afterward sort
undertakings in view of need. At that point the calculation compute the culmination time of each assignment on various services, and calendar each undertaking onto an administration which can finish the errand as quickly as time permits as indicated by the arranged errand line. Be that as it may, in this procedure need can change powerfully an expansion consistently this can take care of the —starvation issue and take after FCFS standard. Trial result accomplishes well execution and load balancing by QoS driving structure both need and finishing time. In 1995 Kennedy and Eberhart [11] Particle Swarm presented The PSO calculation Optimization (PSO) as a meta-heuristics strategy is a self-versatile worldwide pursuit based streamlining system it is indistinguishable to other populace based algorithms like Genetic algorithms (GA) be that as it may, there is no immediate recombination of people of the populace The PSO calculation concentrates on limiting the aggregate cost of calculation of an application work process. As a measure of execution, Authors utilized cost for finish execution of utilization as a metric. The goal is to limit the aggregate cost of execution of utilization work processes on Cloud computing conditions. Results demonstrate that PSO based undertaking asset mapping can accomplish no less than three times cost investment funds when contrasted with Best Resource Selection (BRS) based mapping for our application work process. What's more, PSO adjusts the load on register assets by conveying assignments to accessible assets.

III. Clouds Load Balancing Metrics

Load balancing metrics are considered in cloud computing for evaluating the performance of an algorithm to determine its effectiveness in surviving cloud challenges. Below are the qualitative metrics used in evaluating the performance of the existing cloud load balancing algorithms.

• Response time: This is the amount of time taken between submission of a request and the first responding time that is produced by a load balancing algorithm in a distributed system. However, reduction in waiting time helps in improving the responsiveness of a virtual machines.

• Performance: It is used to determine how effective the system when implementing load balancing.

• Resource utilization: Checking the utilization of resources in load balancing is required to optimize the efficiency of load balancing algorithm.

• Throughput: This is the overall task completion. The throughput required to be high for better performance of system. It comprises of overhead as a result of task migrations from one domain to another.

• Cost overhead: It’s used to find amount of overhead involved when implementing load balancing techniques. It should be acquired at minimum to enable load balancing works effectively.

• Adaptability: When a calculation performs stack adjusting with any finites number of hubs for a framework, is called versatility. Change in adaptability of a calculation is vital to stack adjusting.

• Fault resistance: A calculation in this case can in any case perform stack adjusting notwithstanding when there is a disappointment in one of the hub. A blame tolerant calculation is useful for stack adjusting to change to different hubs that will cover disappointment of the fizzled hubs.

• Migration time: The time taken to relocate figuring assets starting with one specific hub then onto the next ought to be limited to improve the execution of the whole framework.

IV. Proposed Methodology

The Resource designation for stack adjusting is composed on Eucalyptus private cloud. It has the better execution for dynamic load adjusting. Eucalyptus is involved six parts: Cloud Controller, Walrus, Cluster Controller, Storage Controller, Node Controller and a discretionary VMware Broker Other than the VMware Broker, every segment is a remain solitary web benefit. Cloud Controller goes about as a Front-end interface for clients. It knows general status of cloud assets and controllers. Group Controller acts an Administers organizing assets and process hubs.

Walrus Controller Storage for virtual machine pictures to use by figure hubs. Capacity Controller goes about as a relentless stockpiling gadget which
can be mounted inside dynamic virtual machines. Compute Nodes gives the execution condition to virtual machines in the cloud. Load Balancing is performed by Cluster Controller. It goes about as the front end for a bunch inside an Eucalyptus cloud and speaks with the Storage Controller and Node Controller. The CC oversees occasion execution and Service Level Agreements per group.

This proposed engineering comprises of three modules. Initially module is work scheduler. The customer presents their solicitations on server which is available in the IaaS cloud condition. At the point when an accessible asset undertaking is allotted to a cloud, first asset accessibility in this cloud will be checked by work scheduler. An occupation scheduler records the execution timetable of all assets utilizing a space and to check the asset accessibility in this cloud.

Second module is resource allocation. In this module the incoming requests are directed to the virtual machines. Third module is load balancer. Load balancer will allow incoming request to be routed into servers that host the web application.

A. Job scheduler

The client submit their requests on server which is present in the IaaS cloud environment, the requests are placed in the queue and send to the server. In scheduler [5] the requests are equally distributed to the nodes. When an available resource task is assigned to a cloud, first resource availability in this cloud will be checked by the job scheduler. A job scheduler record execution schedule of all resources using a slot. The job scheduler is used to simulate from users for virtual machines. In this module we have considered a web service which has different task and this web service application are placed in the server and made the user to access these applications from the client machine.

B. Resource allocation

In resource allocation we have to select the available resources depending upon the specification of our task by using load balancing condition. When the user sends the request to server, the server identifies [3] the request and redirects to the particular virtual machine. The virtual machine processes the requested task and sends the corresponding response to the client request. If any one of virtual machine have overloaded, apply load balancing policies and we can redirect the incoming request to the other virtual machines to balance the load among cloud environment.

C. Load balancing workflow

In load balancer we have created a web application in which multiple numbers of users is allowed to access the virtual machines efficiently from the client side. Initially the different users use the application and submit the task to the server. The user’s requests are placed in the task queue and it is sent to the load balancer. The load balancer checks for the available virtual resources which are connected to it and it also has the status information of every virtual machine which is connected to it. The status information represents the status of every virtual machine, such as whether it is in busy state or in available state. Based on the status of virtual machines the task can be allocated to the virtual resources. For checking the availability of virtual machine we have to update the status of the virtual machine shown in Figure 3.
The status update is represented in the form of table to represent the status of CPU speed, memory and disk of different virtual machines. The status update is used to represent the availability of the virtual machines whether it is in busy state or in the idle state so that it can be allocated for another task if it is free. The output of this load balancing workflow is to make use of available resource efficiently. When the incoming request is overloaded then the admin set the priority to each request. Multiple numbers of requests are submitted in the server and set the priority to each request. The priority is based on capacity and load factor. For using priority the client requests are redirected to the nodes.

D. Dynamic Priority Based Algorithm

A dynamic load balancing algorithm checks the previous state of a node while distributing the load. The proposed algorithm uses dynamic priority for the requests based on which the virtual machines are scheduled. It schedules the VMs to the requests depending upon their priority value, which varies dynamically based on their load factor. This dynamic priority concept leads to better utilization of the resources. Priority of a request is assigned depending upon its capacity and the load factor.

Algorithm: Priority (PB-SA) Input: N number of client Request Output: Balance the Request

Algorithm check priority schedule
1. Get the available VMList
   //find the appropriate VMList from job scheduler
2. If priority1 is not assign to request
3. Priority1=max available VM
4. Else if priority1 is set
5. Turn ON priority1
6. Used VMList
7. If VM is not used VMList then Add VM to used VMList
8. Deploy request on new VM
9. If priority N is not assign goto step1
10. Assign VM to ClientRequests
11. If assigned the requests then return Successful
12. End for
13. Return available VMList

Priority [1] based scheduling Algorithm is intended to be used by organizations need to implement small to medium sized local clouds. This algorithm should scale to larger sized clouds because one of the main contributions of the cluster controller is load balancing compute nodes.

<table>
<thead>
<tr>
<th>Request</th>
<th>Arrival Time( ms)</th>
<th>Execute Time(m s)</th>
<th>Priority (P)</th>
<th>Service Time(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>R0</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>R1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>R2</td>
<td>2</td>
<td>8</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>R3</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>16</td>
</tr>
</tbody>
</table>

Each process is assigned as a priority. Process with highest priority is to be executed first and so on. Processes with same priority are executed on first come first serve basis. Priority can be decided based on memory, time or any other resource requirement shown in Table.1.
The advantage of using dynamic load balancing is that if any node fails, it will not stop the progress of the system; it will only affect the system performance. When compared to a centralized environment in distributed dynamic load balanced system, the nodes can work together. However, selecting a suitable server needs real time communication with the other nodes of the network and hence, generates multiple numbers of messages in the network. Dynamic load balancer uses the load balancing policies for keeping track of updated information.

V. Conclusion

Cloud Computing technology is increasingly being used in enterprises and business markets. In cloud computing system, an effective resource allocation strategy with Service Level Agreement (SLA) is required to achieve user satisfaction and maximizing the profit for Cloud Service Providers (CSP). This paper summarizes the main types of task scheduling and optimization strategy with its impacts in cloud system. Hence, future research is to come up with a much smarter and secured optimized task to resource allocation algorithms and also a framework to strengthen the entire cloud computing paradigm.

References


