Load Balance Scheduling Algorithm for Serving of Requests in Cloud Networks Using Software Defined Networks

Dr. Chinthagunta Mukundha  
Associate Professor, Dept of IT, Sreenidhi Institute of Science & Technology, Hyderabad, India.

P. Gayathri  
Associate Professor, Dept of CSE, KGR CET, Moinabad, India.

Dr. I. Surya Prabha  
Professor, Department of IT, Institute of Aeronautical Engineering, Hyderabad, India.

Abstract

Present days Cloud Computing becomes a popular computing area for providing the computing resources. Cloud Computing offers the services like Infrastructure, Platform, Software, Communication and Monitoring. Generally Cloud Networks works based on the zones in particular location. If the area is near by the requested client then the then that particular zone serve for the request. Sometimes some of the zones are busy and sometime some zones are idle in use. In That case the resources provided in the cloud zones are not utilized properly. To avoid that problem In this paper we proposed that a concept called Load balancing scheduling algorithm for the cloud networks using Software Defined Networks (SDN). This algorithm works on the basis of equal distribution of client requests to all the zones available to the particular cloud networks. Load balancing algorithm targets to optimize the utilization of the resource by maximizing the throughput, minimizing the response time, and avoiding overloading of any single resource. Software Defined Networks are used to apply the Software functionality to the Hardware devices like switches routers etc. By using Software Defined Networks the algorithm is placed in the cloud routers then the router will decide the zone for the cloud client based on the resource availability in the cloud networks.

Keywords: Communication, Infrastructure, Software Defined Networks, zones, cloud computing, load balancing

Introduction

By considering Cloud networks overhead, techniques for load balancing are of significant importance. Load balancing directly affects the application and service availability for cloud clients. Load balancing algorithm targets to take maximum advantage of the resource by maximizing the throughput, minimizing the response time, and avoiding overloading of any single resource. To avoid heavy-traffic cloud networks flux and reduce the risk on a single server, many data centers are provided dedicated hardware methods to enable load balancing in order to support a large number of users in bulk. The hardware systems that are used in cloud networks are expensive to procure, can be technically challenging to be deployed and it may require human interaction to work consistently. To avoid manual intervention in the operation of cloud networks we introduce a specialized network called software defined networks. Software-defined networking is a type of networking provides an easy, convenient, flexible network flow control method with a nominal investment to reduce cost and increase benefit for large number of cloud clients. It manages data flow by means of software implementation of switches and routers. When a data flow arrives at a switch and router a flow table lookup has to be carried out. Flow tables are widely used in Software Defined Networks. For each network flow, the headers and counters will be revised if flow changes are required or actions are imposed. By recording the header information into a database, an OpenFlow router can process the data flow according to the header records. Based on the Software Defined Network model with a centralized controller, an OpenFlow router is designed for different rules to control the network traffic by organizing the header records. The flow control system will theoretically make it possible to define an algorithm to balance the network load.

This Paper aims at to present a new cloud computing model for dealing load balance in cloud networks by using load balancing scheduling algorithm with software defined networks. The SDN controller can manage all in bound and out bound traffic of each server in cloud networks through routers. By deploying dynamically extensible load balancing Strategy in the SDN routers an efficient model is proposed to reduce the packet delay in traditional communication networks and guarantee the reliability for huge cloud customers, continuity and timeliness of their business. In comparison with existing load balancing algorithm this algorithm solves issues like more cost, low reliability and less extensibility. This technique aims at maximizing the resources of cloud networks available in different zones unlike in traditional cloud networks utilization of cloud networks are not maximize and sometimes it may be idle based on the client requests. Fig 1 shows the architecture of SDN layers with routers as a network devices.
The cloud network offers three service models. First service is Infrastructure as a service model offers computing resources like processing, storage, network bandwidth etc. Second service is Plat form as a Service model offers platform for the developers to host their application for testing. Third Service is Software as a Service model offers applications for the end customers.

Related Work
The main aim of load balancing has always been to efficiently and fairly distribute every computing resource and improve resource utilization over the cloud networks. Many researchers in the past have proposed different scheduling algorithms like static, dynamic and mixed scheduling strategies. Generally in all proposed resource allocation algorithms if the resources is free in nearest zones then the resource is allocated for the cloud client otherwise request is discarded irrespective of other zone availability of the resources. Some load balancing algorithms allocate the resources if available otherwise it puts the requests in a FIFO queue.

In Native Strategy if we have x services in a software workflow and t is the time for execute each software service. A set of n number of virtual computing resources may be allocated the network bandwidth is also allocated based on the requirement of the data transfers between two services. This strategy considers only the stream line execution and it serves as a baseline for the performance.

In a FIFO Strategy an assumption is made that on every computing resource, all the services can be deployed. The scheduler may ask any resource to compute any task. Due to infrastructure, redeployment of services is not necessary. It is considered as an optimal strategy. In this case, the same bandwidth is allocated to all the links in the infrastructure. Due to data transfer time, when the bandwidth is small, the total cost is high, but when the bandwidth increases, then both cost and execution time decrease. The optimization is used to approximate the optimal bandwidth. This strategy only works for the identical resources and bandwidth is not optimized between each pair of resources.

The optimized strategy divides the execution of the workflow into multiple stages and at each stage, bandwidth and resources are allocated independently. And also in every stage, the minimization algorithm is executed to allocate the optimum number of resources for the services involved in the current stage. An algorithm is required to determine the number of stages. Directed execution graphs (DAG) are made up of the workflow of services. The DAG is divided into the execution stages. A special virtual infrastructure executes the execution stages. In every execution stage, the infrastructure is reconfigured for deployment for services involved in the stage. The resources are allocated based on the number of invocations required for each service.

In Autonomous age based load balancing algorithm for dynamic load balancing. It uses three agents; Load agent, Channel agent and Migration agent to define its load balancing scheme. The load and channel agents are static while Migration agents represent an ant. The load agent manages data policy and maintained all details of datacenter and at the same time responsible for calculating the load on every available virtual machine (VM) when new task are allocated in a datacenter. The load agent is supported with a VM_Load Fitness table. The fitness table organizes the list of all details of virtual machines properties in a datacenter such as id, memory, and fitness value and load status of all virtual machines. When the load agent completed the controlling policy, the channel agent controls the transfer policy, selection policy and location policy. Upon request received from the load agents, the channel agent initiates a communication with the migration agents. The migration agent then moved to other datacenters and communicates with load agent of the datacenter to enquire about the status of VMs presents. Upon receiving the status, it communicates to its parent channel agent. This approach reduces service time and overcome the challenge of overloaded virtual machines.

In this hybrid scheduling algorithm based on throttled and equally spread current execution load balancing algorithm. The hybrid algorithm initializes all VMs allocation status to AVAILABLE in the VM state List. With the “hash map” list size available to determine overloaded and under loaded VM. When a data centre collects a new request to be processed, the Datacenter manager queries new load balancer for next allocation. If the “hash map” list size is less than the state list size, a VM is allocated otherwise the load balancer has to wait until the virtual machine gets free. When the request is been processed by a datacenter, and “Datacenter Manager” receives the cloudlet response, it notices the load balancer of the VM de-allocation. Finally, the load balancer updates the status of the VM in VMs state list and “hash map” list. A Cloud Analyst was used to implement the algorithm. The result shows a significant improvement in response time, and the datacenter processing time as well as the cost of processing request compared to the existing load balancing algorithm.

The nature inspired pre-emptive task scheduling for load balancing in cloud networks considered balancing pre-emptive independent task on virtual machines (VMs) based on honey bee foraging behavior. To consider the pre-emption of task based on priority, when virtual machine are over loaded, the task with the highest priority is removed to under loaded virtual machine. Upon arrival of the migrated task to the under
loaded virtual machine, its priority and the required execution time is then compared with the priority and the execution time of the already running task in the under loaded virtual machine. If the migrated task has higher priority and the execution time is less than the already running executing task, then the executing task is paused and pre-empted until migrated task finishes executing, the pre-empted task now return to the state of execution, otherwise, the running task has to finish executing. This approach works better for a static cloud environment where high speed connection is required. At present, many research studies on load based resources scheduling are based upon dynamic migration of services from one request to another for serving more number of client requests. Sandpiper systems carry out hotspot probing and dynamic monitoring on the utility of system’s memory resources, CPU and network bandwidth. It also provides black box and white-box resource monitoring methods. The system's main focus is to determine how to dispose hotspots through the remapping of resources in cloud networks migration and how to define hotspot memory. Load based Distributed Resource Scheduler (DRS) is a tool that balances and distributes computing volume by using available resources in a cloud environment. Thus, using dynamic migration all of the above systems can achieve system load balance; but frequent dynamic migration would employ a large number of resources that might lead to degrading the entire system performance.

The main advantage of load balancing in cloud computing helps to guaranteed that customers needs are achieved at a quick response time, keeping the make span of the system at good state, ensured better throughput, energy conservation and also coordinating management of resource pool across cloud networks. When data centre becomes overload with processing task, the benefit of load balancing migrate task to available data centres that are underutilized for processing. Load balancing helps to boost productivity of cloud datacenter while maintaining service level agreement (SLA). Some benefits of load balancing in cloud computing is to ensure that:

- Resources are available on demand
- Saving of energy if less amount of requests
- Processing time is reduced
- Efficient utilization of resources upon high or low
- Ensures better communication across network
- Low cost

Software-Defined Networking (SDN) is an emerging network architecture where network control is decoupled from forwarding and it is directly programmable Per this definition, SDN is defined by two characteristics, namely decoupling of control and data planes, and programmability on the control plane. Nevertheless, neither of these two signatures of SDN is totally new in network architecture, as detailed in the following. First, several previous efforts have been made to promote network programmability. One example is the concept of active networking that attempts to control a network in a real-time manner using software. Switch Ware is an active networking solution, allowing packets flowing through a network to modify operations of the network dynamically. Similarly, software routing suites on conventional PC hardware, such as Click, XORP, Quagga, and BIRD also attempt to create extensible software routers by making network devices programmable. Behavior of these network devices can be modified by loading different or modifying existing routing software. Second, the spirit of decoupling between control and data planes has been proliferated during the last decade. all of the decision making is accomplished by the centralized SDN controller. SDN facilitates efficient coordination between the cellular and Wi-Fi networks.

**Proposed System**

In this paper we present a technique for load balancing in cloud using software defined networks. By applying software defined networks to the cloud networks we can effectively organize client requests in the cloud computing. Scheduling mechanisms for cloud computing is increased in day to day by cloud popularity. Scheduling is the process of mapping tasks to available resources on the basis of tasks’ characteristics and requirements. It is an essential aspect in efficacious working of cloud as many task parameters need to be considered for proper scheduling. The available resources should be utilized efficiently without affecting the service parameters of cloud. Scheduling process in cloud can be generalized into three stages namely

1. **Discovery of Resources and filtering-SDN router**
   Discover the resources available in the Cloud, if the resources are available then resources are filter based on the client requests.

2. **Selecting Resources-If** target resource is selected based on certain client parameters of task and resource. Then resource is selected for allocation.

3. **Submission of Task-Resource is submitted to the client task based o the client request.**

**Algorithm**

1. Define n is number of request to access the resources in the cloud networks.
2. Define z is number of zones available to a particular cloud networks.
3. Each request is allocated to one zone based on zone availability.
4. If all zones are busy then it checks the zone that contains less number of requests then the request is allocated to the zone having less number of requests.
5. After allocating each request to the zones the Request Allocation Table (RAT) in each router is updated with new allocation details.
6. Then updated Request Allocation Table is forwarded to each router in the network and each router is updated with new data.
7. All the routers are working based on the routing technology in the networking.

Above algorithm describes the total working process of Load balancing algorithm in Software Defined Networks like router and switches in the cloud networks.
system in large-scale networks. The proposed system is the simulation conducted to evaluate the proposed algorithm for cloud networks using software defined networks. This paper presents a new version of load balancing scheduling algorithm for cloud networks using software defined networks with zone availability. This technique improves the performance of cloud networks to process the client requests to allocate the resources in the cloud. Various terminologies and concepts were thoroughly looked and thus explained. This paper will give way to more future findings regarding the scheduling techniques in a cloud environment.

**Conclusion**

In this paper, we present a new version of load balancing scheduling algorithm for cloud networks using software defined networks with zone availability. This technique improves the performance of cloud networks to process the client requests to allocate the resources in the cloud. Various terminologies and concepts were thoroughly looked and thus explained. This paper will give way to more future findings regarding the scheduling techniques in a cloud environment.

**References**


18, no. 6, pp. 803-813, 2013.


