Survey of Load Balancing in Cloud Computing

[1]Yashika Sharma, [2]Anshu Sharma
[1]Research scholar, [2]Assistant Professor,
[1]Department of Computer Science Engineering, RPIIT, Karnal , India

***Abstract* - Cloud computing there are many tasks that needs to be executed by the available resources to acquire high performance, reduce task completion time, minimize response time, utilization of resource usage and etc. Scheduling theory for cloud computing is gaining a lot of attention with increasing popularity in this cloud era. This technology aims to offer distributed, virtualized, and elastic resources as utilities to end users. It has the potential to support full realization of 'computing as a utility' in the near future. With the support of virtualization technology, cloud platforms enable enterprises to lease computing power in the form of virtual machines to users. Because these users may use hundreds of thousands of virtual machines (VMs)[4], it is difficult to manually assign tasks to computing resources in clouds.**

***Index Terms* - NIST,IAAS,PAAS,VMM**

I. INTRODUCTION

Cloud Computing denotes to operating, organizing, and retrieving the services over the internet. It offers online services. Cloud computing is the provision of computing facilities using the Internet. Cloud facilities permit persons and industries to use software and hardware that are accomplished through third revelries at distant locations. Instances of cloud amenities comprise online file storing, social networking sites, webmail, and online commercial presentations [3]. The cloud computing prototypical permits right to use to info and computer resources from wherever that a network association is accessible. Cloud computing delivers a common group of resources, together with data storage space, networks, computer processing power, and dedicated commercial and customer requests.

Cloud computing is an umbrella. The word cast-off to mention to internet based improvement and facilities. The cloud is a representation for the Internet. A number of features describe cloud information, requests amenities and infrastructure [4]:

- Distantlypresented: Facilities or information are presented on somebody else's substructure.
- Omnipresent: Facilities or information are obtainablecommencingeveryplace.
- Commodified: The result is a utility computing model similar to traditional that of traditional utilities, like gas and electricity. You pay for what you would like.

The subsequent meaning of cloud computing has been advanced by the U.S. National Institute of Standards and Technology (NIST):

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models."

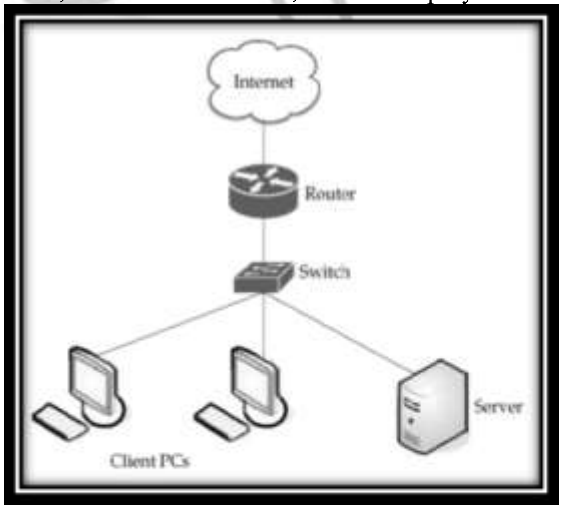

Fig 1.1 Transforming to info using internet

Cloud computing is an emerging device. It carries infrastructure, platform, and software that are reachable as subscription-based amenities in a pay-as-you-go example to clienteles. These facilities are mentioned by way of Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS) in businesses. Cloud computing is internet- based computing.

II.Cloud Computing Service Models

Cloud Computing is a Distributed Computing paradigmwhich provides services to their customers on the basis of asper usage by particular customer. That is, use as much or lessyou want to use, use services when you want to use and payonly for what you have used. Cloud computing is a constructthat allows you to use applications that actually reside on alocation different from your machine location and provides adifferent virtualized platform that helps user to accomplishtheir jobs with minimum completion time and minimumcosts.[8]

- **Infrastructure As A Service:-** :-Infrastructure as a Service (IaaS) is a self-service model for managing remote data center infrastructures. IaaS provides virtualized computing resources over the Internet hosted by a third party such as Amazon Web Services, Microsoft Azure or Google. Instead of an organization purchasing hardware, companies purchase IaaS based on a consumption model. It is like buying electricity. You only pay for what you use. This model enables companies to add, delete or reconfigure IT infrastructure on-demand. Many IT organizations rely on IaaS because they are more familiar with IaaS, especially if they have years of experience with virtual environments or strict security and regulatory requirements that can only be met through IaaS.An example of IaaS is Amazon EC2, GoGrid[6][8].
- **Platform As A Service :-**Platform as a Service (PaaS) allows organizations to build, run and manage applications without the IT infrastructure. This makes it easier and faster to develop, test and deploy applications. Developers can focus on writing code and create applications without worrying about time-consuming IT infrastructure activities such as provisioning servers, storage and backup. PaaS brings more value to cloud. It can reduce your management overhead and lower your costs. PaaS also makes it easier for you to innovate and scale your services on demand. Examples of PaaS are Amazon Web Services, Google App Engine and Microsoft Azure[16]WOLF (cloud middleware)[18].
- **Software As A Service :-**Software as a service (SaaS) replaces the traditional on-device software with software that is licensed on a subscription basis. It is centrally hosted in the cloud. A good example is Salesforce.com. Most SaaS applications can be accessed directly from a web browser without any downloads or installations required. However, some SaaS applications require plug-in. Examples of SaaS are Google Docs, Microsoft Office 365, salesforce.com etc[16] Facebook .

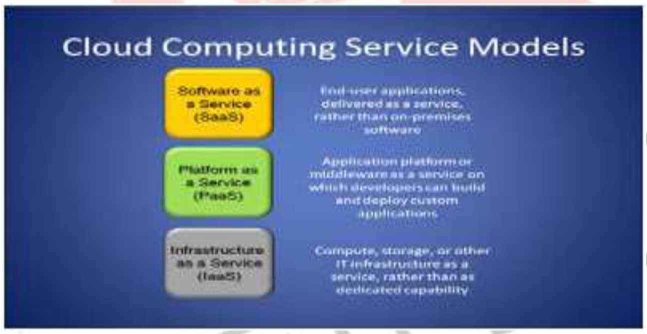
Cloud Computing Service Models
Software as a Service (SaaS)
End-user applications, delivered as a service, rather than on-premises software
Platform as a Service (PaaS)
Application platform or middleware as a service on which developers can build and deploy custom applications
Infrastructure as a Service (IaaS)
Compute, storage, or other IT infrastructure as a service, rather than as dedicated capability

Figure1. Service Models Of Cloud Computing

III.Related Work

In [9]we deal with the various methodologies adopted to handle all the processes and jobs concurrently executing and waiting into the web application and web server
housed into the same system or different systems. Various issues like virtual resources, queuing strategies, resource managers etc. we has been discussed here apart from the main coverage points. Web application remains in contact with the user to work inreal-time with the user's requests to avoid any kind of delay.ame system or different onesthe fact that the level of performance and efficienc level differs by a value, not so large, but, significant enough.Each microsecond of resource' usage counts a lot in terms ofCPU performance and efficiency. h the application and they demand for the subsequent number of resources as needed by them and make sure that they are used in the most appropriate manner.Web application remains in contact with the user to work in real-time with the user's requests to avoid any kind of delay. An application can have any number of processes [4] under it which also depends on the ongoing processes. Different processes are kept in different process groups [4] to a certain order of uniformity in the scheduling [4] process and therefore, are allotted their respective ProcessGroupID. Every process can contribute to any number of jobs depending upon the type and number of operations to be performed by that particular instance of process. Jobs are assigned a particular ID. a detailed working of different job scheduling methods behind the web application on a web server on the same system or on different systems in a virtual cloud computing environment The various terminologies and concepts were thoroughly looked and thus explained. more future findings regarding the scheduling techniques in a cloud environment. More efficient and faster ways to schedule jobs and increase CPU throughput [2, 4] needs to be discovered. Also, this will fuel the greater knowledge and popularity of cloud environment.

In [10]presents a novel heuristic scheduling algorithm, called hyper-heuristic scheduling algorithm (HHSA), to find better scheduling solutions for cloud computing systems. The diversity detection and improvement detection operators are employed by the proposed algorithm to dynamically determine which low-level heuristic is to be used in finding better candidate solutions. To evaluate the performance of the proposed method, this study compares the proposed method with several state-of-the-art scheduling algorithms, by having all of them implemented on CloudSim (a simulator) and Hadoop (a real system). The results show that HHSA can significantly reduce the makespan of task scheduling compared with the other scheduling algorithms evaluated in this paper, on both CloudSim and Hadoop.
In [3] proposal the use of two basic swarm intelligent metaheuristic algorithms, including ant colony optimization and priority based Artificial Bee Colony (ABC). The main objective was to combine some metaheuristic properties of ants and honey bees to prepare a hybrid algorithm for effective scheduling and uniform distribution of workload among the cloud virtual resources. The proposed algorithm was decentralized to avoid the single point of failure. The parameters used were number of processing cores, MIPS searching, memory size of virtual machine and virtual machine bandwidth. In the implemented algorithm, ants were considered as cloudlets that are composed of userbases and food as virtual machine.Cloud Analyst toolkit was used for simulation and performance analysis. The results were derived in the form of statistical metrics. The simulation results demonstrated that the proposed approach outperforms the previous approaches like Round-Robin, Equally Spread Current Execution Load (ESCE) and metaheuristic ACO.
[4]performed the comparative analysis of four load balancing policies/strategies including RR, Throttled algorithms were implemented using ESCE and FCES. All four algorithms were implemented using Cloud analyst simulation tool. The parameters used for comparative analysis were: average response time, data center request servicing time and total cost of data center. The performance of Round Robin algorithm was found to be the best among all.
In [7] Type Aconfiguration referred to traditional hypervisor basedor container based virtualized compute clement. Type B configuration referred to a model that start with a native hypervisors and expand to support multiple virtualized data centers comprised of additional native hypervisors with a data center manager and/or container based OSs with cluster manager. The main focus was on Type c infrastructure model. In this, an unmodified host OS executed a hosted hypervisor such as VMware servers or workstations. Multiple virtualized data center were supported by the hosted hypervisors with virtual cloud environment. A Virtual Computing Lab (VCL, a cloud infrastructure) was used for implementation of these approaches and provisioning custom virtualized cloud environment to cloud consumers.
In [8] implemented a VM fork abstraction i.e. Snowflock. It was designed as an open source project built on the Xen 3.0.3 VMM and implemented as a combination of modification to Xen VMM and daemons that run in domain (). A distributed system is formed of these daemons that control the life cycle of VMs, by orchestrating their cloning and deallocation. Allocation management is supported by snowflock with EGA platform and Sun Grid Engine. A simple internal resource manager was used to track memory and CPU allocations on each physical machine. VM fork enabled the simple implementation deployment of services based on familiar programming patterns that rely on forks ability to quickly instantiate state full worker. Snowflock has broad applicability to other applications as well such as flash crowd handling, execution of untrusted code components, instantaneous testing and many more. The locality of memory access across cloned VMs made it beneficial to distribute VM state using multicast. This instantiation of large number of VMs was possible at low cost similar to that of forking a single copy.

IV. PROPOSED MODEL

4.1 CloudSim

The CloudSim software framework and architectural components. Earlier versions of CloudSim used SimJava discrete event simulation engine [5] as a provider of core functionalities required for higher-level simulation frameworks, such as queuing and processing of events, creation of system components (services, host, data center, broker, virtual machines), communication between components, and management of the simulation clock. Nevertheless, this layer has been removed in order to allow advanced operations not supported by SimJava.
CloudSim provides novel support for modeling and simulation of virtualized Cloud-based data center environments such as dedicated management interfaces for virtual machines (VMs), memory, storage, and bandwidth. CloudSim layer manages the instantiation and execution of core entities (VMs, hosts, data centers, application) during the simulation period. This layer is capable of concurrently instantiating and transparently managing a large scale Cloud infrastructure consisting of thousands of system components. The fundamental issues such as provisioning of hosts to VMs based on user requests, managing application execution, and dynamic monitoring are handled by this layer.
A Cloud provider, who wants to study the efficacy of different policies in allocating its hosts, would need to implement his strategies at this layer by programmatically extending the core VM provisioning functionality. There is a clear distinction at this layer on how a host is allocated to different competing VMs in the Cloud. A Cloud host can be concurrently shared among a number of VMs that execute applications based on user-defined QoS specifications. Similarly, a cloud application developer can evaluate performance and perform some workload profiling by modeling the application characteristics, and considering network characteristics among different data centers and users using features presented in this level.

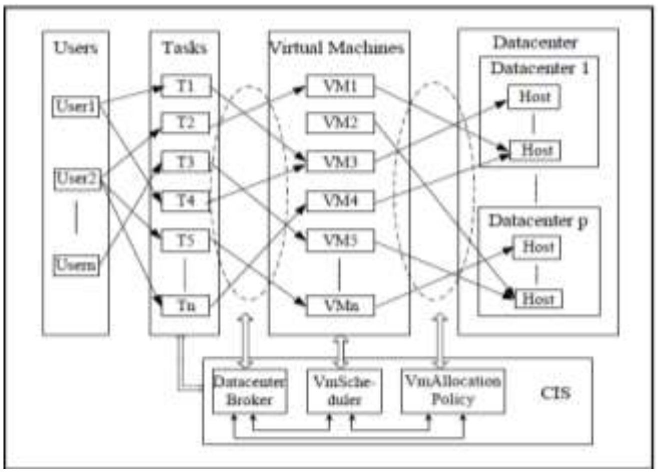

Figure 4.1– Cloud Sim

4.2 CloudCoordinator.
This abstract class provides federation capacity to a data center. It is able to periodically monitor the internal data center status, and based on that it is able to decide whether part of the data center load can be migrated or not. Concrete implementation of this object contains the specific sensors and the policy to apply during load transferring. Monitoring is performed periodically by the *updateDatacenter()* method that queries to different Sensors, one for each monitored feature. Discovery is implemented in the *setDatacenter()*abstract method, which allows for CloudSim users to determine the way discovery should performed in the specific implemented scenario. This component can also be used to implement services such as Amazon’s Load-Balancer. Developer aiming for deploying their application services across multiple clouds can extend this class for implementing their custom resource provisioning policies.

4.3 loudlet
This class models the Cloud-based application services (content delivery, social networking, business workflow),which are commonly deployed in the data centers. CloudSim represents the complexity of an application in terms of itscomputational requirements. Every application component has a pre-assigned instruction length (inherited from GridSim'sGridlet component) and amount of data transfer (both pre and post fetches) that needs to be undertaken for successfullyhosting the application. This class can also be extended to support modeling of other metrics for application execution, suchtransactions in database-oriented applications.

4.4 DataCenter
This class models the core infrastructure level services (hardware, software) offered by service provider(Amazon, Azure, App Engine) in a Cloud computing environment. It encapsulates a set of compute hosts that can be either homogeneous or heterogeneous as regards to their resource configurations (memory, cores, capacity, and storage). Furthermore, every DataCenter component instantiates a generalized application provisioning component that implements a
set of policies for allocating bandwidth, memory, and storage devices.

4.5 DatacenterBroker.
This class models a broker, which is responsible for mediating between users and service providers depending on users’ QoS requirements and deploys service tasks across Clouds. The broker acting on behalf of users. It identifies suitable Cloud service providers through the Cloud Information Service (CIS) and negotiates with them for an allocation of resources that meets QoS needs of users. The researchers and system developers must extend this class for conducting experiments with their custom developed application placement policies. The difference between this component and the CloudCoordinator is that the former represents the customer (i.e, decisions of this components are made in order to increase user-related performance metrics), while the former acts on behalf of the data center, i.e, it tries to maximize performance from the data center point of view, without considering needs of specific
computer.

4.6 DatacenterCharacteristics
This class extends GridSim’sResourceCharacteristics by adding description of cost to use datacenter resources.

4.6.1 DatacenterTags.
This class contains tags to be used along with the messages, to identify different message types.

4.6.2 FederatedDataCenter
This class extends the basic DataCenter by adding federation capacity to the data center. It is aDataCenter having a CloudCoordinator component associated to it. The associated Cloud-Coordinator communicates with other CloudCoordinators when it is necessary and calls methods from the FederatedDataCenter to trigger events related to load redistribution and other activities.

4.7 Host

This class models a physical service in a Cloud-based Data Center. It contains an amount of memory and storage, a list of processing elements (to represent a multi-core machine), an allocation policy for sharing the processing power among virtual machines, and policies to provisioning memory and bandwidth to the virtual machines.

V. Conclusion

In this paper , Cloud computing is becoming one of the most explosively expanding technologies in the computing industry today. It enables users to migrate their data and computation to a remote location with minimal impact on system performance. There are a number of underlying technologies, services, and infrastructure-level configurations that make Cloud computing possible. One of the most important technologies is the use of virtualization.

REFERENCES

[1] Kornai and Arvinder Kaur, "A New Hybrid Scheduling in Cloud Environment," International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, pp. 407-411, 2015.
[2] Kobra Etminani and M. Naghibzadeh, "A Min-Min Max-Min Selective Algorihtm for Grid Task Scheduling," in ICI 2007. 3rd IEEE/IFIP International Conference in Central Asia, 2007.
[3] Gao Ming and Hao Li, "An Improved Algorithm Based on Max-Min for Cloud Task Scheduling," Springer, pp. 217-223, 2012.
[4] T. Kok_ilavani and Dr. D.I. George Amalarethinam, "Load Balanced MM-Min Algorithm for Static Meta-Task Scheduling in Grid Computing," International Journal of Computer Applications, vol. 20, no. 2, pp. 43-49, April 2011.
[5] Sameer Singh Chauhan and R. C. Joshi, "A Weighted Mean Time MM-MM Max-Min Selective Scheduling Strategy for Independent Tasks on Grid," 2nd International Advance Computing Conference, IEEE, pp. 4-9, 2010.
[6] Huankai Chen, Frank Wang, Na Helian, and Gbola Akanmu, "User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing," IEEE, 2013.
[7] Riemina Priyadarsini and L.Arockiam, "Performance Evaluation of Min-Min and Max-Min Algorithms for Job Scheduling in Federated Cloud," International Journal of Computcr Applications, vol. 99, no. 18, pp. 47-54, August 2014.
[8] D.Doreen Hephzibah Miriam and K.S.Easwarakumar, "A Double Min Min Algorithm for Task Metascheduler on Hypercubic P2P Grid Systems," International Journal of Computer Science issues LICS1, vol. 7, no. 4, July 2010.
[9] Broberg J., Goscinskki A. Buyya r., Introduction t Pvt. 2016 to Cloud Computing. New Delhi: Wiley ludia_ Ltd.
[10] Guilherme Galante and Luis Carlos E. de Bona, "A Survey on Cloud Computing. Elasticity," ACM Fifth International Conference on Utility and Cloud Computing, IEEE, [nline]. pp. 263-270, 2012. May).