ISSN NO. 2320-5407

Journal homepage:http://www.journalijar.com

INTERNATIONAL JOURNEL OF ADVANCED RESEARCH

RESEARCH ARTICLE

National Symposium On Emerging Trends In Computing & Informatics, NSETCI 2016, 12th July 2016, Rajagiri School of Engineering & Technology, Cochin, India

EFFECT OF GROUP JOB SCHEDULING IN CLOUD ENVIRONMENT.

***Mathews Abraham, and Kuttyamma a j.**

Department of Information Technology, Rajagiri School of Engineering & Technology.

Manuscript Info

Key words:
Host,
CloudSim,
Data center,

Abstract

This project introduces an effective scheduling algorithm, which attempts to maximize cloud resources utilization, improve the computation ratio, and reduce makespan, overhead and delay in a cloud-based software system. However, virtualized data centers consume huge amounts of energy, which increases the operational costs. To optimize resource usage and reduce energy consumption of Infrastructure as a Service (IaaS) Cloud, it needs a continuous monitoring and consolidation of VMs using live migration and switching idle hosts to the sleep state. Cloudresources plays an important role in performance, cloud computing mainly focuses how to effectively utilizes the cloud resources to obtain the maximum performances.

Introduction:-

Cloud computing is a technology which uses internet and one remote server to maintain data and various applications. Cloud computing provides significant cost effective IT resources as cost on demand IT based on the actual usage of the customer. Due to rapid growth, many companies are unable to handle their IT requirement even after having an in-house datacenter. Cloud services helps to improve IT capabilities without investing large amounts in new datacenters. This technology helps companies with much more efficient computing by centralizing storage, memory, processing and bandwidth. Optimal resource allocation or task scheduling in the cloud should decide optimal number of systems required in the cloud so that the total cost is minimized and the SLA is upheld. Cloud computing is highly dynamic, and hence, resource allocation problems have to be continuously addressed, as servers become available/no available while at the same time the customer demand fluctuates. Cloud service scheduling is categorized at user level and system level . At user level scheduling deals with problems raised by service provision between providers and customers. The system level scheduling handles resource management within datacenter. Datacenter consists of many physical machines. Millions of tasks from users are received; assignment of these tasks to physical machine is done at datacenter.

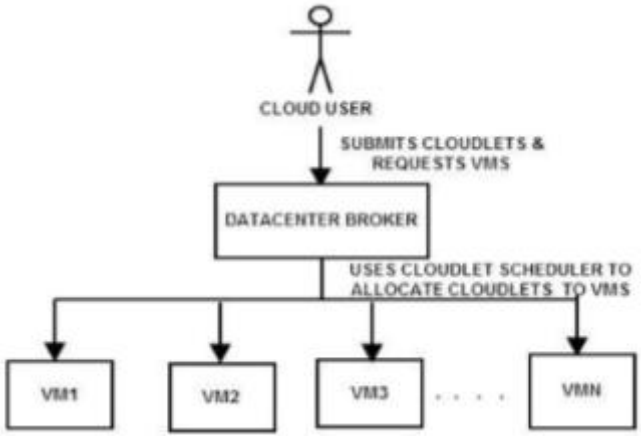
This assignment or scheduling significantly impacts the performance of datacenter. In addition to system utilization, other requirements like QoS, SLA, resource sharing, fault tolerance, reliability, real time satisfaction etc should be taken into consideration. Cloud computing models are shifting. In the cloud/client architecture, the client is a rich application running on an Internet-connected device, and the server is a set of application services hosted in an increasingly elastically scalable cloud computing platform. The cloud is the control point and system or record and applications can span multiple client devices. The client environment may be a native application or browser-based; the increasing power of the browser is available to many client devices, mobile and desktop alike. Robust capabilities in many mobile devices, the increased demand on networks, the cost of networks and the need to manage bandwidth use creates incentives, in some cases, to minimize the cloud application computing and storage footprint, and to exploit the intelligence and storage of the client device. However, the increasingly complex demands of mobile users will drive apps to demand increasing amounts of server-side computing and storage capacity. Resource Allocation is all about integrating cloud provider activities for utilizing and allocating scarce resources within the limit of cloud environment so as to meet the needs of the cloud application. It requires the type

and amount of resources needed by each application in order to complete a user job. The order and time of allocation of resources are also an input for an optimal resource allocation. An important point when allocating resources for incoming requests is how the resources are modeled. There are many levels of abstraction of the services that a cloud can provide for developers, and many parameters that can be optimized during allocation. The modeling and description of the resources should consider at least these requirements in order for the resource allocation works properly. Cloud resources can be seen as any resource (physical or virtual) that developers may request from the Cloud. For example, developers can have network requirements, such as bandwidth and delay, and computational requirements, such as CPU, memory and storage. The development of a suitable resource model and description is the first challenge that a resource allocation must address. An resource allocation also faces the challenge of representing the applications requirements, called resource offering and treatment.

Also, an automatic and dynamic resource allocation must be aware of the current status of the Cloud resources in real time. Thus, mechanisms for resource discovery and monitoring are an essential part of this system. These two mechanisms are also the inputs for optimization algorithms, since it is necessary to know the resources and their status in order to elect those that fulfill all the requirements.

Material and Methods:-

CloudSim is a simulating program from CLOUDS lab in University of Melbourne for cloud computing. It is developed in java platform including the pre-developed modules such as SimJava and GridSim.CloudSim enables seamless modeling, simulation, and experimentation of cloud computing environment and the application services. And it also shows the result of the time, power, and consumption. Cloud computing is a best fit for applications where users have heterogeneous, dynamic, and competing quality of service (QoS) requirements. Different applications have different performance levels, workloads and dynamic application scaling requirements, but these characteristics, service models and deployment models create a vague situation when we use the cloud to host applications. The cloud creates complex provisioning, deployment, and configuration requirements.



Using CloudSim, industry-based developers and researches can focus on specific sys- tem design issues that they want to investigate, rather than considering the lower level details about the cloud-based services and infrastructure. The main components of the CloudSim framework

1) Cloudlet: This class models the cloud based application services such as business workflow, social networking and so on. Cloudlets may also be referred to as tasks.
2) Cloudlet Scheduler: This class implements different allocation policies that determine the sharing of processing power among cloudlets in a virtual machine.
3) Host: This class models a physical resource such as a computer or storage server.it encapsulates information such as amount of memory and storage, processing cores, policies for allocation of memory and bandwidth to virtual machines.
4) Virtual machine: This class models a virtual machine. This virtual machine is created on a cloud Host component.
5) Vm allocation policy: This class determines the policy for allocating VMs to Hosts based on various criteria.
6) Datacenter: This class models the core infrastructure level services offered by the cloud service providers. It encapsulates a set of Hosts that can either be homogeneous or heterogeneous.

7) Service broker: The service broker decides which data center should be selected to provide the services to the requests from the user base.
8) Datacenter broker: This class is responsible for mediating between cloud service providers and users. It identifies the suitable service providers and allocates the resources in such a way that the users QoS requirements are met.

Priority job scheduling:-

The cloud user submits the tasks, the tasks are prioritized. Based on the priority, the tasks are allocated to the virtual machines. The task with highest priority is assigned to the virtual machine with largest processing power.

• Step 1. Cloud users submit Cloudlets, along with priority and request for Virtual machines.
• Step 2.The Cloudlet list and the VM request list are passed on to the DatacenterBroker.
• Step 3. The Cloudlet list is sorted such that the Cloudlet of highest priority appears rest.
• Step 4. The VM request list is sorted in decreasing order of the processing power.
• Step 5. The DatacenterBroker uses CloudletScheduler to schedule Cloudlets.
• Step 6: The cloudlets are allocated to the VMs based on priority.
• Step 7: The Cloudlet with highest priority is assigned to the VM with largest processing power.
• Step 8: The VM list is then passed to VMAllocationPolicy for the placement of VM on the Host.
• Step 9:The Hosts will be created on Datacenters. In cloud computing system, there exists some applications with a great deal of light-weight tasks. Dispatching these fine grained tasks to a pool of resources that provide high processing capability is not economical and consumes extra waiting time and turnaround time by comparing a coarse grained task allocation to the resource . Since the overall turn around time includes each task scheduling time, execution time and transmission time. A large amount of ne grained tasks will spend a lot of time on scheduling and transmission.

It is rather impractical to consume the resource processing capability, and lowers resource utilization rate when a fine grained task is allocated and executed to a resource with high processing capability. The total turnaround time of fine grained tasks can be further reduced by grouping these fine grained tasks as coarse grained tasks in the entire scheduling process.

Group job scheduling:-

Task grouping implies that similar type or characteristic tasks can be grouped together in a group and scheduled corporately . The dynamic grouping strategy mainly pays attention to utilization of resources. Clients submit tasks to the scheduler and the scheduler gains the correlative characters of resources. Then the scheduler selects a specific resource according to some priorities and multi- plies resource MIPS (processing capability) with granularity size. The calculated result implies a resource total MI (processing requirement) during granularity size. Next the scheduler groups tasks of clients by accumulating each of task MI. By comparing the accumulation result (total tasks MI) and a resource total MI, when the accumulation result is over a resource total MI, grouping will be stopped. The last task added to the group should be removed and then the accumulation result subtracts the last added task MI. The final accumulation result is used for a new task MI. Sequentially, a new grouped task is created, whose MI is the final accumulation result with a unique ID.

Finally, the scheduler allocates the new grouped task to the corresponding resource. The grouping process will continue until all the tasks are performed in new groups and allocated to the cloud resources. Then cloud resources execute all these new grouped tasks and send the executed grouped tasks back to the clients after completion of processing tasks. A basic factor that influences job grouping is granularity size. Granularity size is the time when a job is processed at the resource. It is used to account for the number of jobs executed on a specific resource during a particular time. The total number of tasks, processing requirements of tasks, the total number of resources and processing capabilities of resources need to be considered. Therefore, it is very important to determine the value of granularity size to minimize the overhead, cost and waiting time and to maximize the utilization of these resources.

Phases of group job scheduling:-

Phase 1: Initialize input data and available resources. Random distribution function is used to create tasks (cloudlets). These tasks processing requirements (MI) follows Normal Distribution, since actually all the tasks are generated randomly and cannot be predicted. Random function is used to set up tasks (cloudlets) file size and output file size, which belong to Uniform Distribution. Similarly, random function is used to create resources which MIPS and bandwidth follow Uniform Distribution.

Phase 2: Prioritize tasks with different demands and factors. For bandwidth awareness, the resource which network has highest communication and transmission rate is selected to reduce the transmission latency. The scheduler sorts resources based on bandwidth in a descending order. For processing capability awareness, the resource with largest processing capability is chosen firstly to decrease the execution time and minimize the overhead time. The scheduler sorts resources based on processing capabilities in a descending order. The following shows the prioritization method using the sort function.

Phase 3: In the improved task grouping algorithm, coarse grained task is considered and can be executed as an individual grouped task. Then the grouped task is assigned to the sequential resource. The data file size after execution is collected to calculate processing cost. Firstly, justify whether a task MI is less or more than cloud resources MIPS*granularity size. If a task MI is larger than total resMIj, the task is grouped itself as one group. Secondly, otherwise a group is constituted by some fine grained tasks, and maximum task file size and output file size of each group is selected as grouped task file size and grouped task output file size, correspondingly.

Phase 4: Shortest Job First scheduling algorithm is adopted to reduce tasks waiting time. In order to decrease waiting time of tasks, the grouped tasks are sorted based on grouped tasks processing requirements in ascending order. After sorting, the grouped task with shortest processing requirement is executed rest on the corresponding resource, and similarly the grouped task with second shortest processing requirement is operated next resource. The rest can be done in the same approach.

Scheduling mechanisms:-

In the case of cloud computing jobs are allocated to the cloud resources on the basis of how we can effectively utilises the resources to perform the maximum. In the case of improved group job scheduling the user submit set of jobs to the cloud platform then these jobs are allocated to the cloud resources on the basis of division method. In cloud computing environment the number of jobs are divided into two types: jobs with highest mips and jobs with smallest mips.at the same time user can allocate set of priority to the number of jobs so that jobs with highest priority is assigned to the cloud resources, otherwise jobs are allocated on the basis of Mips. After allocation of jobs to the cloud resources set of parameters are used to find out the performance evaluation. Thisinclude makespan,work completion time, energy, power etc. After evaluating this parameters it shows that group job scheduling reduce the job completion time because of jobs are allocated to the cloud resources on the basis of grouping mechanisms so that granularity size and Mips place an important role in job allocation.group job scheduling reduces the amount of work done by the system so that by the use of grouping it will improve the overall performance and utilization of resources

Result and Discussion:-

The study proposes a task grouping scheduling algorithm combined with shortest job rest and band- width awareness algorithms in an attempt to reduce the waiting time and its associated processing costs. Experimental results show utilization of resources and minimized turnaround time, as well as reduced influence on the bottleneck of bandwidth usage. Compared with existing task grouping algorithms, the pro- posed algorithm waiting time has been reduced by 31.38%. Even if in comparison with the task grouping with bandwidth awareness algorithm, the waiting time has also been reduced by 4.63%.The proposed scheduling algorithm achieves a minimum waiting time which is also able to determine deadline constraints. Better yet, the processing time of proposed algorithm has also been reduced to satisfy the real time demand of clients. The empirical results illustrate that the proposed scheduling policies are a significant improvement and serve an important baseline for benchmarking, and for the future development of scheduling algorithms for cloud-based software applications and systems.

Cloud Computing Scheduling is simulated in CloudSim using java programming language. The major benefit of simulation based approach is that it allows cloud developers to test the performance of various scheduling policies in a controllable environment, before these policies are actually deployed in real cloud. This enables the developers to rectify their faults at an earlier stage.

The existing and the proposed algorithms are tested for their efficiency by creating fifteen Virtual Machines which run on six Hosts. Each VM and Host is given specific characteristics like processing capability, storage, bandwidth and so on. The Hosts run on two Datacenters. The algorithms are tested by varying the number of Cloudlets. In the proposed algorithm, each cloudlet is given a specific priority. Based on the priority, the cloudlet will be allocated to the virtual machines. The execution time of the Cloudlets is found using the existing model and the proposed model.

different Type of Job Scheduling Algorithm implemented using cloudsim 3.0.jobs can be allocated to resources effectively in this wayefficient utilization of resources can be performed.

References:-

1) **J. G. Koomey, Growth** in data center electricity use 2005 to 2010, Oakland, CA: Analytics Press. August1.
2) **Darshan Upadhyay**, Scheduler in cloud computing using open source tech-nologies, Int. J. Computer Technology applications, vol 3, 10931098.
3) **Che-Lun Hung, Yu-Chen Hu and Kuan-Ching Li** ,Auto-Scaling Model for Cloud Computing SystemInternational Journal of Hybrid Information Technology ,Vol. 5, No. 2, April, 2012.
4) **Hyejeong Kang ,Jung-in Koh , Yoonhee Kim** ,An Investigation on Scheduling Policies for Cloud-based Software Systems Jia Ru Department of Computing The Hong Kong Polytechnic University Hong Kong SAR
5) **Mohammed, F.C., Dhaliwal, G.S. and Sharma, R.K. (1999).** Clinical efficacy of GnRH analogue (Buserelin) and oestradiol benzoate treatment in anoestrus buffaloes. Indian J. Anim. Sci. 69(5): 310-312.
6) **Mrs.S.Selvarani1, Dr.G.Sudha Sadhasivam.**Improved CostBased Algorithm For Task Scheduling In Cloud Computing
7) **Wanqing You, Kai Qian, and Ying** Qian.Hierachial Queue Based Scheduling .