Research Paper Engineering

Dynamic Allocation of Virtual Machines in Cloud Computing

Ashwini E
Department of computer Science and Engineering, Dr, AIT, Bangalore, India

ABSTRACT

The next generation of computation service will be provided by the cloud computing services. Cloud computing allows business customers to scale up and down their resource usage based on needs. Many of the touted gains in the cloud model come from resource multiplexing through virtualization technology. Dynamic selection of virtual machines plays an important role in providing services to the consumers. This paper discusses the design and implementation of the dispatcher algorithm for effective utilization of the cloud resources. Also presents a case study which examines the implementation of the dispatcher algorithm, by a server, A proper scheduling and efficient load balancing across the network can lead to improve overall system performance and a lower turn-around time for individual tasks

KEYWORDS : Virtual Machine deployment, cloud computing, Resource utilization, Load balancing.

Introduction

Cloud computing as became new horizon for computing environment which aspire to supply an reliable, QoS(Quality of Service) and customized dynamic computing environment for consumers [1]. Parallel computing, grid computing and distributed processing combined together have emerged as cloud computing. The basic belief behind the cloud computing is consumer will not store data in local systems instead it will be stored in remote data centers and accessed through the internet. The IT (Information Technology) industries which provide data storage services will manage and preserve the operations of these data centers. The consumers can retrieve the stored data at any point of time through the cloud service providers using internet connected to a system. In today's business market cloud computing not only provides data storage services but also software and hardware services are also available. These services can be platform as a services (Paas), infrastructure as a service (Iaas) and software as a services (SaaS) [2].

When the consumers request for the resources cloud service provides must provide the resource available by considering the Service Level Agreement (SLA). So in order to make resource available there is a need of efficient and optimized method for scheduling of resources, developing applications on Virtual Machines (VM). Currently, more work is made on scheduling of consumer applications on cloud [3], [4], [5]. Single SLA such as cost of execution and execution time are considered in these approaches. When the consumer request for job execution on the cloud, it usually divided into several tasks. Following research questions are required to be considered when executing this several tasks.

- How to measure the workload of several tasks?
- How to allocate the required resources to execute the several tasks?
- How to schedule and manage the VM's

Typically, efficient provisioning requires two distinct steps or processes: (1) initial static planning step: the initially group the set of VMs, then classify them and deployed onto a set of physical hosts; and (2) dynamic resource provisioning: the allocation of additional resources, creation and migration of VMs, dynamically responds to varying workload. Step 2 runs continuously at production time where in contrast Step 1 is usually performed at the initial system set up time and may only be repeated for overall cleanup and maintenance on a monthly or semi-annually schedule.

In Resource allocation (RA) is the process of allocating available resources to the required consumer application for execution over the internet. If resource allocation is not done properly then it will get waste and there will be a failure in providing a service to the consumers.

Resource Allocation Strategy (RAS) 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 RAS. An optimal RAS should avoid the following criteria as follows:

- Resource contention situation arises when two applications try to access the same resource at the same time.
- Scarcity of resources arises when there are limited resources.
- Resource fragmentation situation arises when the resources are isolated. [There will be enough resources but not able to allocate to the needed application.]
- Over-provisioning of resources arises when the application gets surplus resources than the demanded one.

In section II, we discuss works related to this topic. In section III, models for resource allocation and task scheduling in cloud computing system are presented. We propose our dispatcher algorithms in section IV, followed by experimental result in section V. Finally, we give the conclusion in section VI.

Related Work

In the research work by Jiani at.al [6], actual task execution time and preemptable scheduling is considered for resource allocation. It overcomes the problem of resource contention and increases resource utilization by using different modes of renting computing capacities. But estimating the execution time for a job is a hard task for a user and errors are made very often [7]. But the VM model considered in [6] is heterogeneous and proposed for IaaS.

Dongwan et al. [8] has proposed a decentralized user and virtualized resource management for IaaS by adding a new layer called domain in between the user and the virtualized resources. Based on role based access control (RBAC), virtualized resources are allocated to users through domain layer. Several researchers have developed efficient resource allocations for real time tasks on multiprocessor system. But the studies, scheduled tasks on fixed number of processors. Hence it is lacks in scalability feature of cloud computing [10]. Recent studies on allocating cloud VMs for real time tasks [12], [11], [9] focus on different aspects like infrastructures to enable real-time tasks on VMs and selection of VMs for power management in the data center. But the work by Karthik et al. [10], have allocated the resources based on the speed and cost of different VMs in IaaS. It differs from other related works, by allowing the user to select VMs and reduces cost for the user.

Zhen Kong et al. have discussed mechanism design to allocate virtualized resources among selfish VMs in a non-cooperative cloud environment in [13]. By non-cooperative means, VMs care essentially about their own benefits without any consideration for others. They have utilized stochastic approximation approach to model and analyze QoS performance under various virtual resource allocations. The proposed stochastic resource allocation and management approaches enforced the VMs to report their types truthfully and the virtual resources can be allocated efficiently. The proposed method is very complex and it is not implemented in a practical virtualization cloud system with real workload.

The seminal work of Walsh et al. [14], proposed a general two-layer architecture that uses utility functions, adopted in the context of dynamic and autonomous resource allocation, which consists of local agents and global arbiter. The responsibility of local agents is to calculate utilities, for given current or forecasted workload and range of resources, for each AE and results are transfer to global arbiter. Where, global arbiter computes near-optimal configuration of resources based on the results provided by the local agents. In global arbiter, the new configurations applied by assigning new resources to the AEs and the new configuration computed either at the end of fixed control intervals or in an event triggered manner or anticipated SLA violation.

The dynamic resource allocation based on distributed multiple criteria decisions in computing cloud explain in [15]. In [15] author contribution is tow-fold, first distributed architecture is adopted, in which resource management is divided into independent tasks, each of which is performed by Autonomous Node Agents (NA) in ac cycle of three activities: (1) VM Placement, in it suitable physical machine (PM) is found which is capable of running given VM and then assigned VM to that PM, (2) Monitoring, in it total resources use by hosted VM are monitored by NA, (3) In VM Selection, if local accommodation is not possible, a VM need to migrate at another PM and process loops back to into placement. And second, using PROMETHEE method, NA carry out configuration in parallel through multiple criteria decision analysis. This approach is potentially more feasible in large data centers than centralized approaches.

The problem of resource allocation is considered in [16], to optimize the total profit gained from the multidimensional SLA contracts for multi-tire application. In this the upper bound of total profit is provided with the help of force-directed resource assignment (FRA) heuristic algorithm, in which initial solution is based on provided solution for profit upper bound problem. Next, distribution rates are fixed and local optimization step is use for improving resource sharing. Finally, a resource consolidation technique is applied to consolidate resources to determine the active (ON) servers and further optimize the resource assignment

A solution for dynamic scaling of web application provided in [18] by describing an architecture to scale web application in dynamic manner, based on threshold in a virtualized cloud computing environment. Architecture consists of front-end load balancer, a no. of web application virtual machine. In it apache HTTP Load Balancer is a front-end load-balancer for routing and balancing user requests to web application deployed on Apache HTTP server that are installed in Linux virtual machine. As per the demand these virtual machines are started and provisioned by a provisioning sub-system. But the action of provisioning and de-provisioning of web server virtual machine instances control by a dynamic scaling algorithm based on relevant threshold of web application.

Resource Allocation Model

The overall architecture is as shown in figure 1, According to the characteristics of applications, we propose an algorithm which dispatches the request by referencing CPU computing power**.** The main effort of this dispatching algorithm is to decide which VM to use and creating a new VM's based on resource availability. It is the place where dispatching decisions are made.

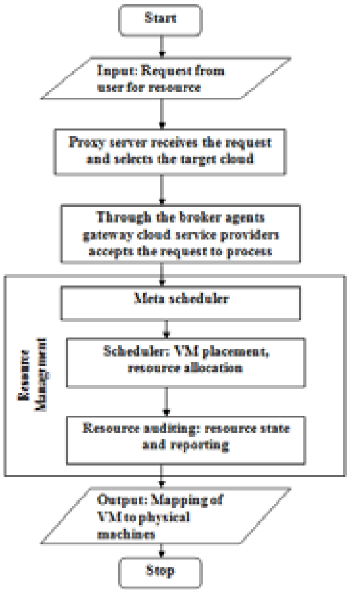
Start
Input: Request from user for resource
Proxy server receives the request and selects the target cloud
Through the broker agents gateway cloud service providers accepts the request to process
Resource Management
Meta scheduler
Scheduler: VM placement, resource allocation
Resource auditing: resource state and reporting
Output: Mapping of VM to physical machines
Stop

Fig1: Resource Management work flow in cloud computing

Once a VM is chosen and the connection is constructed, all remote invocations go through this link are served by this VM. Here we can have the channel objects periodically discard connections in purpose for the reconstruction of connections to less load servers. The network processor records the IP and port information of the client and the selected VM in the VM connection table called VMCT for each constructed connection.

The remote request for resources with the same source IP, the same source port, and the same destination port will be directed to the same destination IP according to VMCT. The response packets from the servers are also directed to the correct VMs by this VM table. The destination port mentioned VMCT is used to identify remoting services Different services distributed and then go to the different ports in our customized consumer channel objects. Dispatching algorithm is to find the least load server for dispatching. Different scheduling methods can be plugged in for this step. In the following, we propose a method to schedule tasks to the server minimizing the estimated task time.

Dynamic Dispacher Algorithm

Load balancing is a technique to enhance resources, utilizing parallelism, exploiting throughput improvisation, and to cut response time through an appropriate distribution of the applications. To minimize the decision time is one of the objectives for load balancing which has yet not been achieved. Proper task scheduling is the only efficient way to guarantee that submitted task are completed reliably and efficiently in case of process failure, processor failure, node crash, network failure, system performance degradation, communication delay, addition of new machines dynamically even though a resource failure occurs which changes the distributed environment [4]. Generally, load balancing mechanisms can be broadly categorized as centralized or decentralized, dynamic or static, and periodic or non-periodic.

```
Algorithm: Dynamic Dispatcher
Input: Get the list of work load and several jobs to execute
Measure the work load
List of available VM's
Initialize Used VM's
If Resource required <= Used VM then
After executing clear the Used VM's
Else
For VM in List of available VM's
Do
Dynamically calculates the required resources and
Dispatches the resources by
Deploying the VM's
Done
```

Output: Mapping of VMs to Physical Machines

Load balancing policies

Load balancing algorithms can be based on many policies; some important policies are defined below [17].

Information policy: This policy specifies what workload information should be collected, when it is to be collected and from where.

Triggering policy: This policy determines the appropriate period to start a load balancing operation.

Resource type policy: This policy classifies a resource as server or receiver of tasks according to its availability status.

Location policy: This policy uses the results of the resource type policy to find a suitable partner for a server or receiver.

Selection policy: This policy defines the tasks that should be migrated from overloaded resources (source) to most idle resources (receiver).

The main objective of load balancing methods is to speed up the execution of applications on resources whose workload varies at run time in unpredictable way. Hence it is significant to define metrics to measure the resource workload. Every dynamic load balancing method must estimate the timely workload information of each resource .

Experimental Results

The experiment is conducted to perform 20 tasks distributed in three different scenarios, first is consumer request to image blurring, the server reads the image and waits for the checking the resource availability. As soon as the server receives text message to blur, the server looks for a VMs utilization and time required will be calculated. Then server dispatches the task on run time to volunteer one or two depends on the RAM resource utilization. Second scenario is consumer request to watermark image on image, the server reads the target image and waits for the image. Third scenario is consumer request to find the number of occurrences of word in a set of files. The experimental result is as shown in Table 1.

VM	Time taken (ms) to perform Image blurring	Time taken (ms) to perform Watermark image with image	Time taken (ms) to perform find No. occurrences of Word in a set of files
1	0.17145	0.15318	0.0434275
2	0.02139	0.05847	0.125624
3	0.01146	0.04638	0.215413
4	0.03215	0.02353	0.135512
5	0.02113	0.01264	0.163517
6	0.01124	0.03125	0.112412

Table 1: Time taken to complete the tasks by each VM's

Conclusion

This paper describes aspects of cloud computing and introduces numerous concepts which illustrate its grand capabilities. Cloud Computing is definitely a promising tendency to solve high demanding applications and its related problems. Main objective of the cloud computing environment is to balance load and achieve high performance. Dynamic nature and complexity of network make load balancing very complex and vulnerable to faults. To maintain entire load of nodes is very hard due to dynamic nature of resources in a network environment. There are a number of factors, which can affect the server performance like load balancing, heterogeneity of resources and resource sharing in the network environment. It focuses on load balancing and presents factors due to which load balancing is initiated, compares existing load balancing algorithms and finally proposes an efficient dispatcher algorithm for network environment.

References

1. Lizhewang, JieTao, Kunze M., Castellanos, A.C,Kramer, D.,Karl,w, "High Performance Computing and Communications", IEEE International Conference HPC-C,2008,pp.825-830.
2. ZhixiongChen,JongP.Yoon,"International Conference on P2P, Parallel,Grid,Cloud and Internet Computing",2010 IEEE:pp 250-257
3. S. K. Garg, R. Buyya, and H. J. Siegel, "Time and cost trade off management for scheduling parallel applications on utility grids," Future Generation. Computer System, 26(8):1344–1355, 2010.
4. S. Pandey, L. Wu, S. M. Guru, and R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," in AINA '10: Proceedings of the 2010, 24th IEEE International Conference on Advanced Information Networking and Applications, pages 400–407, Washington, DC, USA, 2010, IEEE Computer Society.
5. M. Salehi and R. Buyya, "Adapting market-oriented scheduling policies for cloud computing," In Algorithms and Architectures for Parallel Processing, volume 6081 of Lecture Notes in Computer Science, pages 351–362. Springer Berlin / Heidelberg, 2010.
6. Jiyani et al.: Adaptive resource allocation for preemptable jobs in cloud systems (IEEE, 2010), pp.31-36.
7. ShikhareshMajumdar: Resource Management on cloud : Handling uncertainties in Parameters and Policies (CSI communicatons,2011,edn)pp.16-19.
8. Dongwan Shin and HakanAkkan :Domain- based virtualized resource management in cloud computing.
9. K.H Kim et al. Power-aware provisioning of cloud resources for real time services. In international workshop on Middlleware for grids and clouds and e-science, pages 1-6, 2009.
10. Karthik Kumar et al.: Resource Allocation for real time tasks using cloud computing (IEEE, 2011).
11. Shuo Liu Gang Quan ShangpingRen On –Line scheduling of real time services for cloud computing. In world congress on services, pages 459-464, 2010.
12. Wei-Tek Tsai Qihong Shao Xin Sun Elston, J. Service-oriented cloud computing. In world congress on services, pages 473-478, 2010.
13. Zhen Kong et.al : Mechanism Design for Stochastic Virtual Resource Allocation in Non-Cooperative Cloud Systems: 2011 IEEE 4th International Conference on Cloud Computing :pp,614-621.
14. W. E. Walsh, G. Tesauro, J. O. Kephart, and R. Das, "Utility Functions in Autonomic Systems," in ICAC '04: Proceedings of the First International Conference on Autonomic Computing. IEEE Computer Society, pp. 70–77, 2004.
15. Yazir Y.O., Matthews C., Farahbod R., Neville S., Guitouni A., Ganti S., Coady Y., "Dynamic resource allocation based on distributed multiple criteria decisions in computing cloud," in 3rd International Conference on Cloud Computing, Aug. 2010, pp.91-98.
16. Goudarzi H., Pedram M., "Multi-dimensional SLA-based Resource Allocation for Multi-tier Cloud Computing Systems," in IEEEInternational Conference on Cloud Computing, Sep. 2011, pp. 324-331.
17. .Kai Lu, Riky Subrata and Albert Y. Zomaya, Networks & Systems Lab, School of Information Technologies, University of Sydney "An Efficient Load Balancing Algorithm for Heterogeneous Grid Systems Considering Desirability of Grid Sites".
18. Chieu T.C., Mohindra A., Karve A.A., Segal A., "Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment," in IEEE International Conference on e-Business Engineering, Dec. 2009, pp.281-286