Resource Provisioning and Management for IaaS Providers in Cloud Computing

Ajeena Beegom A S
Department of Computer Sc. and Engg.
College of Engineering
Trivandrum
India

M S Rajasree
IIITM-K
Trivandrum
India

ABSTRACT

Cloud computing refers to Internet based distributed computing where the physical resources are pooled at one end and users across the globe can have access to unlimited resources as pay-as-you-go utility computing model. Cloud service users requests computing resources and cloud service provider provides them as virtual machine (VM) instances. The problem addressed here is dynamic VM creation and allocation which benefit users in terms of response time and Cloud Service Providers (CSP) in terms of reduced energy and management cost by increasing the utilization of physical resources which are powered up for the time being and reduce the number of machines which need to be turned on. The proposed system include a demand forecast module which helps provisioning sub system, that manages the dynamic provisioning, in VM creation and management decisions.

General Terms:

Cloud Computing, Resource management

Keywords:

Cloud Computing, Resource Provisioning, IaaS, Resource Allocation Policy

1. INTRODUCTION

Cloud computing implies a service oriented architecture, reduced information technology overhead for the end user, great flexibility, reduced total cost of ownership and on-demand services [1]. It typically involves the provisioning of dynamically scalable and often virtual resources as a service over the Internet. In cloud computing, there exists two categories of users, cloud service providers (CSP) and cloud users. A cloud user is one who requests services and CSPs provide the requested service to the cloud users through the Internet. Depending on the services offered to cloud users, CSPs may be providing infrastructure as a service (IaaS), platform as a service (PaaS) or software as a service (SaaS). For any CSP, the physical resources which form the infrastructure of the cloud need to be provisioned dynamically to suit customer needs. In this environment, end users can arrive and leave at any time, CSP should be able to scale up or down its data centres on multiple criteria such as the delay of virtual resource setup, the migration of existing processes and the resource utilization [3].

The key element in the cloud computing system is virtualization, which allow multiple virtual machines to run concurrently, in limited number of physical machines. To achieve maximum economies of scale, we need flexible and efficient provisioning and scaling system that fully utilizes the underlying physical hosts and adjusts to changing workload demands. These economies include higher profit rate for CSPs, satisfaction at the end user level and energy efficiency and green computing at the environment side. These issues have attracted many researchers into the field of resource provisioning and allocation strategies that suits the requirements of cloud computing systems.

In this work, we aim to develop an algorithm that supports VM provisioning and VM scaling subsystems on when and where to allocate new VMs, when to terminate an allocated VM and informs these systems on the availability of resources for scaling and provisioning, with the help of load prediction subsystem.

2. RELATED WORK

A parallel data processing framework for dynamic resource provisioning in cloud is presented in [4]. The system named Nephele, is introduced as an alternative to MapReduce framework [5], which claims to incorporate the dynamic needs of the computing job and dynamic provisioning of VMs. A gossip protocol for dynamic resource management in large cloud environments is given in [6]. Here, a decentralized design is proposed for dynamic scalability, adaptability and to achieve maximum fairness among compute resources but at the cost of performance overhead. Authors of [7] propose a resource allocation strategy using over booking, advanced reservation, just-in-time bidding and using substitute providers for service delivery. They have done an economy analysis of the proposed method to find the revenue / profit earned by the CSP. Performance of this algorithm heavily depends on over booking and advanced reservation strategies, which incurs more cost on off-peak hours. A request partitioning approach based on iterated local search is employed in [8]. The method include a two phase, splitting of user request among eligible CSPs is done in the first phase and then a mapping of the requested virtual resource to physical mapping is performed in the second phase. The proposed method can be used for inter-cloud scheduling. Authors of [9] uses particle swarm optimization technique to solve resource allocation prob-

lem. They are using non domination principle to find the optimal schedule for the modelled multi-objective optimization problem, which is a heuristic approach to solve the same. In [10], honey bees are used for revenue optimal dynamic allocation in Internet server colonies. The authors of [11] has done a detailed analysis on various aspect of energy-aware resource allocation heuristics for efficient management of data centres in cloud. They have proposed an architecture for green cloud computing and put forth many open issues and challenges in this domain and proposed an algorithm for energy efficient management of cloud computing environments. Some of the researchers use predictive modelling to predict future resource requirements based on past data. One such work is proposed by Andreolin et al.[12], which uses workload prediction techniques to cope up with dynamic changes in Internet based systems. The authors propose a two step approach that aims to get a representative view of the load trend from measured raw data and then applies a load prediction algorithm. This approach is suitable to support different decision systems for highly variable contexts. Tammaro et. al.[3] has proposed a few algorithms for resource allocation among clusters for allocating a particular request, but does not deal the allocations problem within a cluster. In [2], a centralised as well as a distributed architecture for cloud is presented and are compared. Our proposed system conceives some idea from this work to define the cloud computing model.

3. CLOUD COMPUTING MODEL

Users of IaaS type of cloud system will be requesting for resources that are needed for their computation / storage purpose. (E.g. - Amazon EC2). Assume the IaaS provider can provide virtual machines (VM) of different capacity range such as small, medium and large VMs. Now at any instance, there will be a set of cloud users requesting for a set of cloud resources of given capacity. Since the cost model depend on the type of VM and the time of usage, the request from any user will contain the type of VM and the time needed. For this work, we have assumed that there exist an IaaS Provider who can rent compute resources (small, medium, large). The provider provides small type of VMs in a set of physical machines, say, 4 Ghz computing power for small types, medium types of VMs in 16 Ghz and large VMs in 32 Ghz.
Further we assume that there exist a set of Server groups (Processors 1 to P) and each can serve any kind of VMs at any point of time, limited to a maximum of *Max* VMs. At any instance of time t, some set of requests will be available and some will already be allocated. Resource allocation problem can be viewed as on-line Bin packing problem where there exists a set bins (Processors , P) and a set of items (user requests, n) where P¡¡n, it is needed to pack the items to bins within a threshold time, but should try to optimize the operating cost. This work aims at predicting the availability of VMs or net capacity available at servers for hosting new VMs at the service provider's end. It will analyze the requests from all the pending users at time frame t for determining the feasibility of allocation within threshold time T in the currently active processors, else it will initiate the process of activating another processor. During feasibility test, the predictor analyzes the current allocation to see whether any of the currently active VM will be released within a time frame. If yes, that capacity will be reused for allocation rather than turning a new server on.

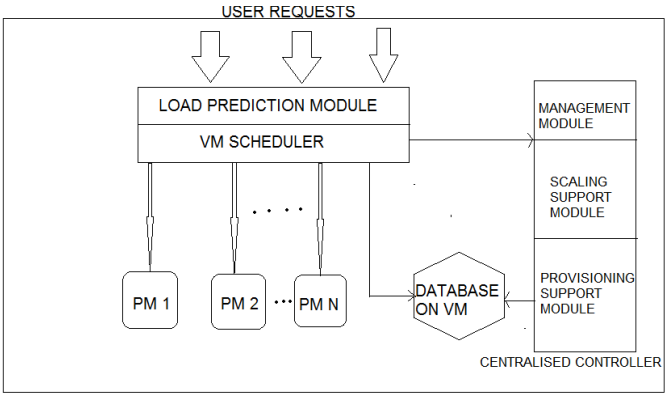

Fig. 1. Proposed Architecture

4. PROPOSED SYSTEM ARCHITECTURE

4.1 Assumptions regarding Architecture

Assume a centralized controller exist with the IaaS provider that can:

(1) Accept user requests, from any where, and analyse the requests
(2) Issue instructions to create VMs in the active processor whenever needed
(3) Issue instructions to activate new processor and new VMs in it whenever needed.
(4) Release a VM and can direct for cleanup action, whenever needed.
(5) Identify VMs of different users uniquely.

The architecture of the proposed system is shown in Fig 1. It consists of a management module, a scaling and provisioning subsystem, database on VM allocation history, a load prediction module and a VM-user mapper(VM scheduler). The VM scheduler schedules the VM onto the available physical machines, details of its allocation such as type, time of creation and requested period, server id and user id are available in the database. On user request, load prediction module analyses the database and inform the management module regarding status of available servers which are powered on. Scaling module becomes active, if it is needed to turn on new physical machines in the server cluster. Provisioning support module issues commands to create / release VMs and the details such as user id, VM creation time, period of activity requested, type of VM and server id in which the VM is instantiated are updated in the database.

5. ALGORITHM

INPUT: User requests at time frame **t** where each request for **j** type VM for duration**Ti** within threshold time **T'**.
Assumptions :

—**P** processors are available during time frame **t**.
—Each processor can have any no. of any type of VM, limited to a maximum of *Vmax*
—If a user request different types of VMs, allocation for all VM types will be done together

OUTPUT : ALLOCATION DECISION BASED ON FCFS POLICY

(1) Check request queue using FCFS policy to see whether the request can be served in the active processors by creating new VMs of requested type. If yes, create new VMs within time **T'** and allocate . Go to STEP 5.
(2) Check to see whether any of the VMs will be released on expiry within **T'**.
(3) If yes, create and allocate VMs in that server group. Goto STEP 5.
(4) Initiate the process of activating (turn on) new server in data centre and create VMs of the requested type.
(5) Repeat Step 1 - 4 until all requests are served.
(6) Stop.

The above algorithm tries to check the availability of server capacity to host the requested service of VM creation and tries to predict whether the request can be served in the currently available servers, if not issue directions to power on another server to host the request without waiting for **T'**.

6. IMPLEMENTATION

The algorithm has been implemented in C++ using a randomly generated request set of 100 requests per time frame, assuming a server pool of 20 which can host small, medium and large type of VMs. Assumptions regarding the threshold time **T'** is fixed based on trial and error method. Algorithm need to be run periodically, after every **t** seconds.

7. RESULTS AND DISCUSSION

The algorithm outputs decisions regarding allocation for each request such as which server group the request fit into or clean up needed in server group at time $T" \leq T'$ when a prior VM completes its execution and suggestions regarding turning on a new server or shutting down an up-server. This decision is made available to the scheduler block which schedules the VMs accordingly. Efforts are made to have the running and management cost of servers to optimal, since always the requests will be served in an up-server if the capacity is available or will be available within **T'**. We have done the simulation studies based on twenty servers which are up for the time being where each can at the most handle sixteen VMs of different types. Based on the availability of processing and computing power, small, medium and large type VMs are instantiated in the servers. We have simulated based on the assumption that a set of VMs are alive and their expected life time is known. For a new request of VM instantiation, the algorithm checks in the available servers to see whether the request can be considered without turning on any other servers within a threshold period, else steps will be taken to instantiate the new VM in a new server. From the experimental results, it is seen that the algorithm tries to utilize the full capacity of an up server to its maximum, thus contributing much to power savings in data centres.

7.1 Analysis

The algorithm employs sequential search in the server group for finding availability of servers and greedy approach is employed for allocation. Also sequential processing of the incoming request queue is done to serve each request (FCFS policy). If there are *n* request in a time frame *t* and if there are *k* server groups, then each request may take at most *k* scans to find a decision, and the running time of the algorithm is *O (nk)*. Additional VM setting up / release time is involved for each request. As the value of *n* and *k* increases, the algorithm takes much time to converge.

7.2 Future work

The algorithm employs greedy approach, when ever it finds a server of sufficient capacity, now or in the near future, it allots that server to the given request. This may cause some of the servers overloaded since the search is sequential. As the running time is more for sequential search algorithms, heuristic search may be employed for better performance. Statistical load estimation and prediction techniques may be incorporated to analyze past behaviour of the systems and make provisioning decisions accordingly. Also the algorithm assume that there are no faulty servers which may not be true in a real scenario. Algorithms may be developed to address these issues.

8. CONCLUSION

The importance of cloud computing for scientific as well as business applications increase day by day. A good VM provisioning strategy is of utmost importance in this scenario to benefit end user as well as the Cloud Service Provider. A sequential search algorithm is proposed in this work to decide on VM provisioning decision in the cloud scenario. The cloud model uses a centralised approach to resource allocation, predicting the appropriate server to host VM request. The algorithm is cost optimal at the cost of O(nk) running time.

9. REFERENCES

[1] Mladen A Vouck, *Cloud Computing - Issues, Research and Implementation*,Journal of Computing and Information Technology, Vol -16, 2008.

[2] T. Chieu and H. Chan, *Dynamic resource allocation via distributed decisions in cloud environment*, IEEE eighth International Conference on e-Business Engineering (ICEBE), pp 125-130, 2011.

[3] D Tammaro, E A Doumith, S A Zahr, J Smets and M Gagnaire, *Dynamic resource allocation in cloud environment under time variant job requests*, Proc. of IEEE third International Conference on cloud computing technology and science, pp 592-598, 2011.

[4] Daniel Warneke, *Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud*, IEEE Transactions on Parallel and Distributed Systems, pp 985-987, 2011.

[5] J. Dean and S. Ghemawat, *MapReduce: Simplified Data Processing on Large Clusters*, Proceedings of Sixth Conference Symposium on Operating Systems Design and Implementation (OSDI 04), pp 10, 2004.

[6] F. Wuhib, R. Stadler and M. Spreitzer, *A gossip protocol for dynamic resource management in large cloud environments*, IEEE transactions on network and service management vol 9, pp 213 - 225, 2012.

[7] Kyle Chard and Kris Bubendorfer, *High performance resource allocation strategies for computational economies*, IEEE Transactions on Parallel and Distributed Systems,2012.

[8] A. Leivadeas, C. Papagianni and A. Papavassiliou, *Efficient resource mapping framework over networked clouds via iterated local search based request partitioning*, IEEE Transactions on Parallel and Distributed Systems-Special edition on Cloud Computing, 2012.

[9] M. Feng, X. Wang, Y. Zhang and J. Li, *Multi-objective particle swarm optimization for resource allocation in cloud computing*, IEEE International conference, CCIS2012, 2012

[10] S. Nakrani, C. Tovey, *On honey bees and dynamic allocation in an Internet Server Colony*, 2nd International workshop on mathematics and algorithms of social insects, CCIS2012, 2003

[11] A. Beloglazov, J. Abawajy, R. Buyya, *Energy-aware resource allocation heuristics for efficient management of data centres for cloud computing*, Elsevier journal of Future generation computer systems, 2012

[12] M. Andreolini, S. Casolari, *Load prediction models in web based systems*, ACM conference on performance evaluation methodologies and tools, 2006