Chapter 1

Introduction

Cloud computing systems have received considerable growing interest recently as their potential applications are based on the client-server paradigm running on the user's machine, while the computations are accomplished on the cloud. Many cloud applications are data-intensive and employ a number of illustrations which run multiple instances of the service and require reliable and an in-order delivery of messages. It is an extension of distributed computing, parallel computing and grid computing which renders secure, immediate, convenient data storage and computing power with the assistance of internet. Cloud computing is abstracting and outsourcing hardware or software resources across the internet, frequently used by a third party on a pay-as-you-go basis. One of the primary motivations for cloud computing is the chance to deploy and use software systems without initial IT investments, instead outsourcing infrastructure to third parties and paying only variable costs associated with the actual resources consumed.

In cloud computing systems (e.g., Amazon's EC2 [1] and Rackspace cloud [2]) dynamically scalable and often virtualized resources are provided as a service over the internet. There are two types of players involved in the system: cloud providers and cloud users. Cloud providers hold large computing resources in their massive data-centers and rent them to the users on a pay-per-usage basis. The cloud users having tasks with unpredictable loads lease the resources from the providers to run their applications. The interaction between the two players are shown in figure 1.1. On receiving the request from the cloud users, the providers look for the resources to satisfy the user requests and assign the required resources, typically in the form of virtual machines (VMs). When a user completes the task with the assigned resources, it returns the resources and pays on a pay-per-usage basis to the provider.

One difficult aspect of cloud system is that the players involved in the system are often different parties having their own business interest. The providers want to maximize the generation of revenue with minimum investment by squeezing their computing resources; for example, by hosting as many VMs as possible on each physical machines. In other words, cloud providers want to maximize the resource utilization. From the users point of view, it may frustrate the cloud user as placing of too many VMs on a single physical machine may cause the performance degradation. Thus, to maintain service quality, the providers may eject existing VMs or reject resource requests, which could make the environment even more unpredictable. The optimal resource allocation is more complex as both parties do not share information with each other. For example, providers do not exhibit the configuration of machines and networks as such information is vital to their business. Similarly, users do not expose the details of their workload, such as source codes and data sets to others, including the providers.

In addition, with the advance of technology the data-centers as well as the pool of resources of cloud providers are becoming heterogeneous and comprise of diverse equipments. Thus, there is a need to make the cloud services and cloud-oriented applications efficient so that appropriate resources can be allocated at a right time to an application for utilization of the resources effectively.

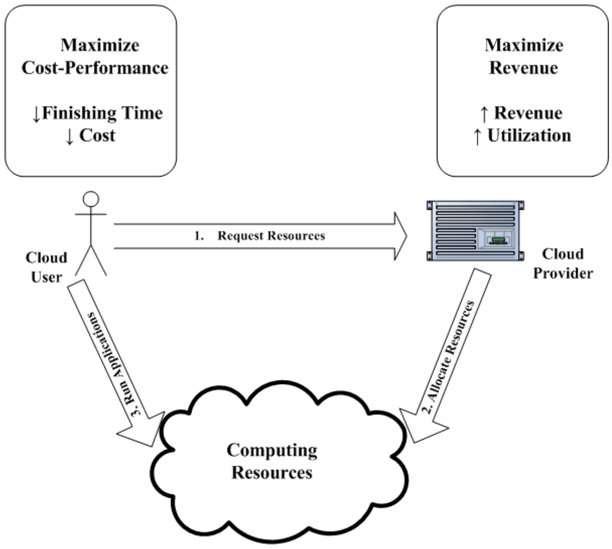

Figure 1.1: Cloud Usage Scenario

1.1 Scheduling and Performance Issues

The resource scheduling and management are the core functions of any man-made system. Performance, functionality and cost are the basic components which affect the core functions and are the important criteria for the evaluation of the system. An inefficient resource scheduling has a direct negative impact on the performance and the cost of the system and an indirect effect on the functionality of the system. A cloud is a complex interconnected network system with a very large number of shared resources with unpredictable client requests and affected by external events. So, cloud resource management requires complex policies and decisions for multi-objective optimization. It is difficult because of the complexity of the system and of the unpredictable client requests in some peak hours.

The resource management strategies associated with the three cloud delivery models, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) are different and in all the cases the service providers are facing with the difficulty of large fluctuating loads which challenge the claim of cloud elasticity. SaaS gives the users potentiality to use applications rendered by the service provider but sets aside no control of the platform or the infrastructure. PaaS devotes the capability to deploy consumer-created or gained applications employing programming languages and tools supported by the provider. IaaS permits the user to deploy and operate arbitrary software, which can include operating systems and applications. In some cases, if the spike can be statistically computed in advance, the resources can be provisioned in advance to maintain the elasticity of the system. For an unplanned spike, auto-scaling can be used provided that: (a) there is a dynamic allocation and deallocation of resources from available pool of resources and (b) there is reallocation of resources through some monitoring system. Auto-scaling is supported by PaaS services, such as Google App Engine. Auto-scaling for IaaS is complicated due to the lack of standards. Autonomic policies are of great interest due to the scale of the system, the large number of service requests, the large user population, and the unpredictability of the load.

1.2 Motivation for our work

Distributed computing has been widely used for many years and is concerned with running the software concurrently on multiple machines by using the client/server architecture. Grid computing is a form of distributed computing which allows a large number of internet-worked computers to work as a virtual super computer providing extensive computational resources to perform a very complex task. Though grid computing has been widely used in academics for more than two decades, in practice it has a very little use. In the recent past, organizations were purchasing their own computing resources and taking total responsibility for the whole maintenance of the infrastructure resulting in additional expenses.

Cloud computing [1, 3–5] developed on many of the combined principles of grid computing, virtualization and service oriented architecture, has the ability to supply resources on the pay-per-usage basic on the clients' demand over the internet. One of the primary motivations for using cloud computing is to use the software systems without initial IT investments, may be outsourcing infrastructure from third parties and paying only the variable costs associated with the actual resources consumed.

The most important requirement for a cloud computing environment is furnishing reliable quality of service (QoS) [6]. It can be defined in terms of minimal throughput, maximal response time or latency delivered by the deployed system. The Service Level Agreements (SLA) between the client and the service provider describe such characteristics [7]. Aggressive consolidation and variability of the workload may cause a problem that some VMs may not get the required amount of resources when requested. This may cause performance loss in terms of increased response time, or failures in the worst case. So the cloud providers have to meet the QoS parameters and maintain the performance measures while minimizing consumption of energy. Therefore, there is necessity to device methods to measure the performance of the system to help the cloud provider when to provision and migrate the VMs. With this motivation, we identify the major goals of the thesis.

1.3 Goals of our work

The primary goal of our work is to measure the performance of the virtual machines (VMs) in different environments to help the cloud providers for utilizing the VMs efficiently. To address this broad objective, we identify the following goals:

- To propose an analytical queueing model for performance management on cloud which can analyze the need to dynamically create and remove virtual machines in order to implement scaling up and down.

- To measure the performances under various load, network time-delay and buffer sizes and optimize the QoS parameters through flexible resource utilization in finite population cloud environment. This provisioning technique may help the cloud operators to tune the resources accordingly to match the offerings with requirements.

- To develop an analytical model for the management of different arrival modes in cloud computing based on service level agreement.
- To investigate the performance measures in dynamic provisioning and resource management for cloud based multi-tier applications.
- To develop an efficient model to process bulk requests concurrently in cloud computing.

1.4 Organization of the Thesis

The rest of the thesis is organized into chapters as follows.
Chapter 2 provides a gist of the background theory. For the sake of conciseness, we do not aim to present an exhaustive treatment of the background theory. Instead, we provide a brief introduction aimed at highlighting the basic concepts. These basic concepts and definitions are used in the subsequent chapters of this thesis.

Chapter 3 provides a queueing based model for performance management on a cloud. This model has been applied to analyze the need to dynamically create and remove virtual machines in order to implement scaling up and down. We measured the probabilities of jth VM being busy which analyzes the creation, removal and live placement of jth VM in the system.

Chapter 4 presents two service policies along with an analytical resource prediction model for each policy for private cloud systems. Various performance measures under various load, network time-delay and buffer sizes of both the systems have been measured. This provisioning techniques may help the cloud operators to tune the resources accordingly to match the offerings with requirements.

Chapter 5 provides the various system measures when two categories of client requests are received at the cloud center, one may be from the premier subscribers and other from the ordinary subscribers. To minimize the total expected cost, a cost model has been developed which determines the optimal number of VMs and the optimal system capacity.

Chapter 6 presents the data center architecture for multi-tier cloud applications. An efficient utilization of resource model has been discussed which dynamically increases the mean service rate of the VMs to avoid congestion in multi-tier cloud environment.

Chapter 7 provides a new cloud resource reservation model with a bulk service mechanism that exploits the capacities of network links to deliver bulk content efficiently.

Chapter 8 concludes the thesis with a summary of our contributions. We also briefly discuss the possible future directions for research work.